



TUGAS AKHIR - TE 145561

**Sistem Monitoring dan Notifikasi pada Prototipe
KVARH Meter Berbasis *Internet of Things* (IoT)**

Novi Ainur Riza
NRP. 2214038008

Dosen Pembimbing
Ir. Sjamsjul Anam, MT.
Fajar Budiman, ST., M.Sc.

DEPARTEMEN TEKNIK ELEKTRO OTOMASI
Fakultas Vokasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



FINAL PROJECT - TE 145561

***Monitoring and Notification System on KVARH
Meter Prototype Based on Internet of Things (IoT)***

Novi Ainur Riza
NRP. 2214038008

Supervisor
Ir. Sjamsjul Anam, MT.
Fajar Budiman, ST., M.Sc.

*DEPARTEMENT OF ELECTRICAL ENGINEERING AUTOMATION
Faculty of Vokasi
Sepuluh Nopember Insitute of Technology
Surabaya 2017*

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things (IoT)***” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

**SISTEM MONITORING DAN NOTIFIKASI PADA PROTOTIPE
KVARH METER BERBASIS INTERNET OF THINGS (IOT)**

TUGAS AKHIR

Diajukan Guna Memenuhi sebagian Persyaratan
untuk Memperoleh Gelar Ahli Madya Teknik
Pada

Bidang Studi Teknik Listrik
Departemen Teknik Elektro Otomasi
Institut Teknologi Sepuluh Nopember



Menyetujui,
Dosen Pembimbing 1

Samsul Anam, MT.
NIP. 19630725-199003 1 002

Menyetujui,
Dosen Pembimbing 2

Fajar Budiman, ST., M.Sc.
NIP. 19860707 201404 1 001

**SURABAYA
JULI, 2017**

-----Halaman ini sengaja dikosongkan-----

SISTEM MONITORING DAN NOTIFIKASI PADA PROTOTIPE KVARH METER BERBASIS INTERNET OF THINGS (IOT)

Nama : Novi Ainur Riza
Pembimbing : Ir. Sjamsjul Anam, MT.
Fajar Budiman, ST., M.Sc.

ABSTRAK

Perhitungan biaya kelistrikan merupakan hal yang penting terutama dalam operasional perusahaan. Kesalahan dalam perhitungan biaya kelistrikan dapat berakibat fatal dalam keuangan dan juga operasional perusahaan. Ketidakakuratan dalam perhitungan biaya listrik antara lain disebabkan oleh belum adanya instrumen yang memadai untuk menghitung beban listrik yang dipakai oleh pelanggan serta informasi dari instrumen diterima secara lambat. Permasalahan di atas yang mendasari diciptakannya karya berupa Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT).

Pembuatan Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT) adalah sebagai berikut. Tahap pertama adalah perancangan sistem pada Arduino Mega yang terdiri dari pengambilan data sensor dan pengolahan data. Tahap kedua adalah perancangan *Internet of Things* (IoT), *platform* IoT yang digunakan adalah Ubidots. Pada tahap ini prototipe KVARH meter dihubungkan dengan Ubidots menggunakan Ethernet Shield. Tahap selanjutnya dilakukan pengaturan kondisi pengiriman notifikasi di dalam Ubidots.

Pada pengujian sistem monitoring dan notifikasi didapat data pengiriman data ke Ubidots dan keberhasilan pengiriman notifikasi. Pengiriman data ke Ubidots tingkat ketelitian bilangan desimalnya 4-5 angka di belakang koma. Sedangkan pada serial monitor angka ketelitian bilangan desimalnya hanya 2 angka di belakang koma. Keberhasilan pengiriman notifikasi melalui SMS dan *e-mail* 100%.

Kata Kunci : KVARH Meter, Monitoring, Notifikasi, *Internet of Things* (IoT), Ubidots, SMS, *E-mail*.

-----Halaman ini sengaja dikosongkan-----

**KVARH METER PROTOTYPE MONITORING AND
NOTIFICATION SYSTEM BASED ON INTERNET OF THINGS
(IOT)**

Name : Novi Ainur Riza
Supervisor : Ir. Sjamsjul Anam, MT.
Fajar Budiman, ST., M.Sc.

ABSTRACT

Measurement of electricity cost is an important thing in industrial operations. Failure to measure the cost of electricity could have major effect both economically and operationally. Inaccuracies in electrical cost calculation are caused by unavailability of suitable instrument for the calculation, and also information transfer speed that is relatively slow. Problem mentioned above inspired the writer to make a final project about KVARH Meter Monitoring and Notification System Prototype Based on Internet of Things.

Design of KVARH Meter Monitoring and Notification System Prototype Based on Internet of Things was as follows. First step was to design of Arduino Mega system that consists of sensor data measuring and data processing. Second step was to designing Internet of Things. Ubidots was used as IoT platform. In this step KVARH meter prototype connected to Ubidots using Ethernet Shield. Next step is adjustment of notification sending adjustment in Ubidots to SMS and E-mail

Ubidots data transmission log and notification transmission success data had been obtained from notification and monitoring system test. Ubidots transmission log data resolution is 4-5 decimals number behind comma, where monitoring serial number resolution is 2 numbers behind comma. Success rate of notification transmission from SMS and e-mail is 100%

Keywords : KVARH Meter, Monitoring, Notification, Internet of Things (IoT), Ubidots, SMS, E-mail.

-----Halaman ini sengaja dikosongkan-----

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT yang selalu memberikan rahmat dan hidayah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik. Shalawat serta salam semoga selalu dilimpahkan kepada Rasulullah Muhammad SAW, keluarga, sahabat, dan umat muslim yang senantiasa meneladani beliau.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma 3 pada Bidang Studi Teknik Listrik, Departemen Teknik Elektro Otomasi, Fakultas Vokasi, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT)

Penulis mengucapkan terima kasih kepada beberapa pihak yang telah membantu dalam penyusunan dan penyelesaian dari Tugas Akhir ini, diantaranya:

1. Kedua orang tua yang selalu memberikan dukungan, semangat, dan doa kepada penulis.
2. Bapak Ir. Sjamsjul Anam, MT. dan Bapak Fajar Budiman, ST., M.Sc. selaku Dosen Pembimbing atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini.
3. Anggota kelompok Tugas Akhir atas bantuan dan kerjasama yang telah diberikan.
4. Keluarga besar Andromeda 2014 yang selalu memberikan bantuan yang tidak terkira terhadap penulis.

Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, Juli 2017

Penulis

-----Halaman ini sengaja dikosongkan-----

DAFTAR ISI

HALAMAN JUDUL	i
PERNYATAAN KEASLIAN TUGAS AKHIR	v
HALAMAN PENGESAHAN	vii
ABSTRAK	ix
<i>ABSTRACT</i>	xi
KATA PENGANTAR	xiii
DAFTAR ISI	xv
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xix
 BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Permasalahan	2
1.3 Batasan Masalah	2
1.4 Tujuan	3
1.5 Sistematika Laporan	3
1.6 Relevansi	4
 BAB II TEORI DASAR	
2.1 KVARH Meter	5
2.2 Daya Listrik	5
2.2.1 Daya Kompleks	5
2.2.2 Daya Aktif	6
2.2.3 Daya Reaktif	6
2.2.4 Segitiga Daya	7
2.2.5 Faktor Daya	7
2.3 <i>Board</i> Arduino Mega	8
2.4 Arduino IDE	10
2.5 Sensor Arus SCT013	11
2.6 Sensor Tegangan	11
2.7 <i>Real Time Clock</i> DS1307 Modules (RTC)	12
2.8 <i>Liquid Crystal Display Keypad</i> (LCD Keypad)	13
2.9 <i>Module Data Logger</i>	14
2.10 <i>Miniature Circuit Breaker</i> (MCB)	15
2.11 Ethernet Shield Arduino Module	16
2.12 IOT Platform - Ubidots API	17

BAB III	PERANCANGAN SISTEM	
3.1	Gambaran Umum Sistem	19
3.2	Perancangan Mekanik	20
3.3	Perancangan <i>Hardware</i>	22
3.3.1	Perancangan Sensor Arus	22
3.3.2	Perancangan Sensor Tegangan	23
3.3.3	Perancangan <i>Wiring Keypad</i> LCD	24
3.3.4	Perancangan <i>Wiring</i> RTC	25
3.3.5	Perancangan <i>Wiring</i> Data Logger	26
3.3.6	<i>Ethernet Shield</i> Arduino	26
3.4	Perancangan <i>Software</i>	27
3.4.1	Perancangan Sistem pada Arduino Mega	27
3.4.2	Perancangan <i>Internet of Things</i> (IoT)	29
BAB IV	PENGUJIAN DAN ANALISA	
4.1	Pengujian <i>Ethernet Shield</i>	39
4.2	Pengujian Arduino Mega 2560	42
4.3	Pengujian <i>Internet of Things</i> (IoT)	45
4.3.1	Pengujian Pengiriman Data ke Ubidots	45
4.3.2	Pengujian Pengiriman Notifikasi	47
4.4	Analisa Relevansi	49
BAB V	PENUTUP	
5.1	Kesimpulan	51
5.2	Saran	52
	DAFTAR PUSTAKA	53
	LAMPIRAN A DATASHEET	
A.1	<i>Datasheet</i> Arduino Mega 2560	A-1
A.2	<i>Datasheet Ethernet Shield</i>	A-5
	LAMPIRAN B LISTING PROGRAM	
B.1	Program pada Arduino Mega	B-1
B.2	Program IP Adress	B-5
	LAMPIRAN C FOTO ALAT	
C.1	Foto Alat	C-1
C.2	Tampilan Ubidots	C-3
	DAFTAR RIWAYAT HIDUP	D-1

DAFTAR GAMBAR

Gambar 2.1	Segitiga Daya	7
Gambar 2.2	<i>Board</i> Arduino Mega	9
Gambar 2.3	Jendela Arduino IDE	10
Gambar 2.4	Sensor Arus SCT013	11
Gambar 2.5	RTC Tiny I2C <i>Modules</i>	13
Gambar 2.6	<i>Liquid Crystal Display Keypad</i> (LCD <i>Keypad</i>)	13
Gambar 2.7	<i>Wiring</i> Modul <i>Micro SD Adapter</i> dengan Arduino Mega	15
Gambar 2.8	<i>Mini Circuit Breaker</i> (MCB)	15
Gambar 2.9	Modul Ethernet Shield	17
Gambar 2.10	Tampilan Ubidots	18
Gambar 3.1	Diagram Fungsional Sistem	19
Gambar 3.2	Perancangan Tata Letak Alat Tigkatan Satu	20
Gambar 3.3	Perancangan Tata Letak Alat Tigkatan Dua	21
Gambar 3.4	Desain Bentuk Depan KVARH Meter	22
Gambar 3.5	Rangkaian Pengkondisional Sensor Arus.....	23
Gambar 3.6	Perancangan Sensor Tegangan	24
Gambar 3.7	Perancangan <i>Wiring</i> LCD Keypad pada Arduino Mega	25
Gambar 3.8	<i>Wiring</i> RTC pada Arduino Mega	25
Gambar 3.9	<i>Wiring</i> Data <i>Logger</i>	26
Gambar 3.10	Skematik Ethernet	27
Gambar 3.11	<i>Flowchart</i> Pemrograman <i>Software</i> Arduino IDE.....	28
Gambar 3.12	Tampilan Ubidots	29
Gambar 3.13	Flowchart Alur Tampilan Ubidots	30
Gambar 3.14	Tampilan Sign Up/Log In Ubidots	31
Gambar 3.15	Tampilan <i>Devices</i> Ubidots	31
Gambar 3.16	Tampilan <i>Variable</i> Ubidots	32
Gambar 3.17	Tampilan Token Ubidots	32
Gambar 3.18	Tampilan <i>Device</i> Ubidots	33
Gambar 3.19	Tampilan Menu Bar Ubidots	33
Gambar 3.20	Tampilan <i>IF THAN Logic</i> Ubidots	34
Gambar 3.21	Tampilan Pengiriman Ubidots	35
Gambar 3.22	Tampilan <i>Event</i> Pengiriman SMS Ubidots	35
Gambar 3.23	Tampilan Pengiriman E-mail Ubidots	36
Gambar 3.24	Tampilan <i>Event</i> Pengiriman E-mail Ubidots	37
Gambar 4.1	Pengujian Besar Nilai Ping pada Jaringan Internet Jarak 1 meter	40
Gambar 4.2	Pengujian Besar Nilai Ping pada Jaringan Internet	

	Jarak 20 Meter	41
Gambar 4.3	Flowchart Pemograman Pengujian pada Pin Arduino	43
Gambar 4.4	Tampilan Dashboard pada Ubidots	45
Gambar 4.5	Tampilan Data pada Serial Monitor	46
Gambar 4.6	Tampilan Notifikasi Melalui SMS	48
Gambar 4.7	Tampilan Notifikasi Melalui e-mail	49

DAFTAR TABEL

Tabel 2.1	Konfigurasi Pin LCD <i>Keypad</i> 16x2	14
Tabel 2.2	Spesifikasi Ethernet Arduino	16
Tabel 4.1	Pengujian Ping pada Jarak 1 Meter	40
Tabel 4.2	Pengujian Ping pada Jarak 10 Meter	40
Tabel 4.3	Pengujian Pin Ethernet	42
Tabel 4.4	Pengujian Pin Arduino Kondisi Logika 1	43
Tabel 4.5	Pengujian Pin Arduino Kondisi Logika 0	44
Tabel 4.6	Data yang Terkirim pada Ubidots	46
Tabel 4.7	Data pada Serial Monitor	47

-----Halaman ini sengaja dikosongkan-----

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perhitungan biaya kelistrikan merupakan hal yang penting terutama dalam operasional perusahaan. Kesalahan dalam perhitungan biaya kelistrikan dapat berakibat fatal dalam keuangan dan juga operasional perusahaan. Ketidakakuratan dalam perhitungan biaya listrik antara lain disebabkan oleh belum adanya instrumen yang memadai untuk menghitung beban listrik yang dipakai oleh pelanggan serta informasi dari instrumen diterima secara lambat.

Dalam sistem tenaga listrik ada daya aktif dan daya reaktif. Daya aktif adalah daya yang harus dibangkitkan di sisi pembangkit dan disalurkan melalui saluran transmisi dan distribusi menuju konsumen, dan dipakai untuk menjalankan peralatan industri. Satuan dari daya aktif adalah watt (W). Sedangkan daya reaktif adalah suatu besaran yang menunjukkan adanya fluktuasi daya di saluran transmisi dan distribusi akibat digunakannya peralatan listrik yang bersifat induktif (misal: motor listrik, trafo dan las listrik). Walaupun namanya adalah daya, daya reaktif ini tidak nyata dan tidak bisa dimanfaatkan. Akan tetapi adanya daya reaktif menyebabkan aliran daya aktif tidak bisa dilakukan secara efisien dan memerlukan peralatan listrik yang kapasitasnya lebih besar dari daya aktif yang diperlukan. Satuan dari daya reaktif adalah VAR (Volt Ampere Reaktif). Untuk menunjukkan seberapa efisien daya aktif disalurkan, dalam teknik tenaga listrik dikenal suatu besaran yang disebut faktor daya. Nilai maksimum faktor daya adalah satu dan nilai minimumnya adalah nol. Semakin tinggi faktor daya maka semakin efisien penyaluran dayanya. [1]

Salah satu yang mempengaruhi biaya listrik adalah nilai KVARH yang dialami oleh pelanggan. KVARH ditentukan oleh PLN untuk pelanggan tertentu sesuai daya berlangganan. KVARH didapat dari energi reaktif atau daya yang tidak terpakai. Pelanggan akan dikenakan denda apabila mengalami kelebihan KVARH, yaitu jika faktor dayanya kurang dari 0,85 setiap bulan, yang menyebabkan nilai KVARH tinggi. Faktor daya yang buruk dapat merugikan pelanggan. Untuk saat ini pelanggan menggunakan *capacitor bank* sebagai solusi perbaikan faktor daya. *Capacitor bank* adalah sekumpulan beberapa kapasitor yang disambung secara paralel untuk mendapatkan kapasitas kapasitif tertentu. Namun dalam hal ini *capacitor bank* tidak dapat memonitoring

dengan efisien. Selain itu alat ini tidak dapat memberikan peringatan kepada pelanggan ketika sudah melewati batas yang ditentukan.

Permasalahan di atas yang mendasari diciptakannya karya berupa Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT). Dengan adanya alat ini dapat memonitoring beban listrik yang dipakai oleh pelanggan. Pelanggan akan diberikan notifikasi apabila faktor daya melebihi batas. Dengan ini pelanggan dapat mengantisipasi agar nilai faktor daya tidak melebihi batas yang sudah ditentukan. Sistem ini menggunakan mikrokontroler Arduino yang dilengkapi dengan sensor arus, sensor tegangan, RTC, LCD, *Module Data Logger*, Relay, MCB, Ethernet Shield dan *Internet of Things* (IoT).

Dengan memaksimalkan penggunaan dan monitoring listrik dengan baik dapat meningkatkan produktifitas konsumen. Dengan demikian Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT), diharapkan dapat menyelesaikan permasalahan yang selama ini dialami oleh perusahaan mengenai biaya kelistrikannya.

1.2 Permasalahan

Beberapa permasalahan yang akan diselesaikan pada Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT) antara lain:

- a. Merancang prototipe KVARH meter pada jaringan tegangan menengah.
- b. Merancang sistem komunikasi yang menghubungkan prototipe KVARH meter dengan *Internet of Things* (IoT).
- c. Merancang sistem pada *Internet of Things* (IoT) untuk monitoring dan notifikasi pada pelanggan.

1.3 Batasan Masalah

Dari perumusan masalah di atas, ada beberapa hal yang perlu dibatasi sehingga Tugas Akhir yang dilakukan dapat tercapai. Batasan masalah dalam pengerjaan Tugas Akhir ini, yaitu:

- a. Tugas Akhir ini dikerjakan secara berkelompok dengan kesepakatan perbedaan titik kerja anggota kelompok.
- b. Sistem komunikasi menggunakan modul Ethernet Shield.
- c. *Interface* sistem dan notifikasi menggunakan *Internet of Things* (IoT), *platform IoT* yang digunakan adalah Ubidots.

- d. Jaringan server komunikasi dianggap dalam kondisi baik dan normal.

1.4 Tujuan

Tujuan dari Tugas Akhir ini adalah:

- a. Membuat sistem monitoring energi listrik pada pelanggan menggunakan *Internet of Things* (IoT) sebagai *interfacing*.
- b. Membuat sistem notifikasi apabila faktor daya kurang dari batas yang sudah ditentukan.
- c. Mempermudah pelanggan dalam menghitung nilai KVARH agar pelanggan dapat mengontrol tagihan kelebihan KVARH.

1.5 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

Bab I Pendahuluan

Bab ini meliputi latar belakang, permasalahan, batasan masalah, tujuan penelitian, sistematika laporan, dan relevansi.

Bab II Teori Dasar

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari KVARH meter, Arduino Mega, Arduino Ide, sensor arus, sensor tegangan, RTC, LCD, *Module Data Logger*, *Relay*, MCB, Ethernet Shield dan *Internet of Things* (IoT).

Bab III Perancangan dan Pembuatan Sistem

Bab ini membahas desain dan perancangan sistem *hardware* maupun *software* pada Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT) berdasarkan teori dasar pada Bab II.

Bab IV Pengujian dan Analisa

Bab ini memuat hasil pengujian serta analisa dari hasil pengujian tersebut

Bab V Penutup

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

1.6 Relevansi

Hasil yang diperoleh dari Tugas Akhir ini diharapkan mampu menyelesaikan permasalahan pada pelanggan, karena adanya Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT) ini pelanggan dapat memonitoring dan mengontrol penggunaan energi listriknya. Serta dapat memberikan fasilitas kepada pelanggan berupa notifikasi apabila faktor daya kurang dari batas yang sudah ditentukan. Sehingga pelanggan dapat mengontrol tagihan kelebihan KVARH oleh PLN.

BAB II

TEORI DASAR

2.1 KVARH Meter

KVARH meter (meter energi reaktif) adalah instrumen ukur listrik integrasi yang mengukur energi reaktif dalam satuan VAR/jam atau kelipatannya yang sesuai, alat ukur ini biasa disebut sebagai alat ukur watt jam. Dalam penggunaannya KVARH meter lebih sering digunakan di dunia industri karena nilai faktor daya sangat berpengaruh terhadap penggunaan energi listrik. KVARH meter terdiri dari dua macam, yaitu:

a. Mekanik

Sistem kerjanya hampir sama dengan KWH meter yaitu perhitungannya berdasarkan perputaran cakram.

b. Digital

Biasanya alat ukur ini menjadi satu dengan KVAH meter, KWH meter, serta cos phi meter.

2.2 Daya Listrik

Daya listrik didefinisikan sebagai laju hantaran energi listrik dalam sirkuit listrik. Terdapat tiga macam daya listrik yang digunakan untuk menggambarkan penggunaan energi listrik, yaitu daya aktif, daya reaktif serta daya kompleks. Dalam sistem listrik AC atau arus bolak-balik ada tiga jenis daya yang dikenal, yaitu:

a. Daya Kompleks

b. Daya Aktif

c. Daya Reaktif

2.2.1 Daya Kompleks

Daya kompleks atau lebih sering dikenal sebagai daya semu adalah penjumlahan secara vektor antara daya aktif dan daya reaktif, dimana:

$$S = P + jQ \dots\dots\dots(2.1)$$

Daya kompleks merupakan daya listrik yang melalui suatu penghantar transmisi atau distribusi. Daya ini merupakan hasil perkalian antara tegangan dan arus yang melalui penghantar.

$$\text{Untuk 1 fasa: } S = V_{L-L} \times I_L \dots\dots\dots(2.2)$$

$$\text{Untuk 3 fasa: } S = \sqrt{3} \times V_{L-L} \times I_L \dots\dots\dots(2.3)$$

S = Daya kompleks (VA)
 V_{L-L} = Tegangan antar saluran (Volt)
 I_L = Arus saluran (Ampere)

2.2.2 Daya Aktif

Daya aktif (daya nyata) merupakan daya listrik yang digunakan untuk keperluan menggerakkan mesin-mesin listrik atau peralatan lainnya. Daya nyata atau daya aktif adalah daya listrik yang digunakan secara nyata, misalnya untuk menghasilkan panas, cahaya atau putaran pada motor listrik. Daya nyata dihasilkan oleh beban-beban listrik yang bersifat resistif murni. Daya aktif ini merupakan pembentukan dari besar tegangan yang kemudian dikalikan dengan besaran arus dan faktor dayanya.

$$\text{Untuk 1 fasa: } P = V_{L-L} \times I_L \times \cos\varphi \dots\dots\dots(2.4)$$

$$\text{Untuk 3 fasa: } P = \sqrt{3} \times V_{L-L} \times I_L \times \cos\varphi \dots\dots\dots(2.5)$$

P = Daya nyata (Watt)
 V_{L-L} = Tegangan antar saluran (Volt)
 I_L = Arus saluran (Ampere)
 $\cos\varphi$ = Faktor daya (standar PLN 0,85)

2.2.3 Daya Reaktif

Daya reaktif dinyatakan dengan satuan VAR (Volt Ampere Reaktan) adalah daya listrik yang dihasilkan oleh beban-beban yang bersifat reaktansi. Daya reaktif merupakan selisih antara daya kompleks yang masuk pada penghantar dengan daya aktif pada penghantar itu sendiri, dimana daya ini terpakai untuk daya mekanik dan panas. Daya reaktif ini adalah hasil kali antara besarnya arus dan tegangan yang dipengaruhi oleh faktor daya.

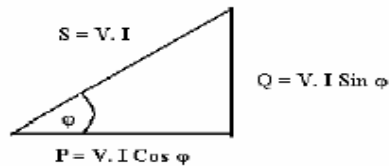
$$\text{Untuk 1 fasa: } Q = V_{L-L} \times I_L \times \sin\varphi \dots\dots\dots(2.6)$$

$$\text{Untuk 3 fasa: } Q = \sqrt{3} \times V_{L-L} \times I_L \times \sin\varphi \dots\dots\dots(2.7)$$

Q	= Daya reaktif (VAR)
V_{L-L}	= Tegangan antar saluran (Volt)
I_L	= Arus saluran (Ampere)
$\sin\phi$	= Faktor Daya (tergantung nilai ϕ)

2.2.4 Segitiga Daya

Dari bermacam daya diatas maka daya listrik digambarkan sebagai segitiga siku, yang secara vektoris adalah penjumlahan daya aktif dan reaktif dan sebagai resultannya adalah daya kompleks. Segitiga daya ditunjukkan pada Gambar 2.1.



Gambar 2.1 Segitiga Daya

2.2.5 Faktor Daya

Faktor daya atau faktor kerja adalah perbandingan antara daya aktif (Watt) dengan daya kompleks/daya total (VA), atau cosinus sudut antara daya aktif dan daya kompleks/daya total. Daya reaktif yang tinggi akan meningkatkan sudut ini dan sebagai hasilnya faktor daya akan menjadi lebih rendah. Faktor daya selalu lebih kecil atau sama dengan satu.

Secara teoritis, jika seluruh beban daya yang dipasok oleh perusahaan listrik memiliki faktor daya satu, maka daya maksimum yang ditransfer setara dengan kapasitas sistem pendistribusian. Sehingga, dengan beban yang terinduksi dan jika faktor daya berkisar dari 0,2 hingga 0,5, maka kapasitas jaringan distribusi listrik menjadi tertekan. Jadi, daya reaktif (VAR) harus serendah mungkin untuk keluaran KW yang sama dalam rangka meminimalkan kebutuhan daya total (VA).

Faktor Daya/Faktor kerja menggambarkan sudut fasa antara daya aktif dan daya kompleks. Faktor daya yang rendah merugikan karena mengakibatkan arus beban tinggi. Perbaikan faktor daya ini menggunakan kapasitor.

Dalam sistem tenaga listrik dikenal 3 jenis faktor daya yaitu faktor daya *unity*, faktor daya terbelakang (*lagging*) dan faktor daya terdahulu (*leading*) yang ditentukan oleh jenis beban yang ada pada sistem.

2.3 Board Arduino Mega

Board Arduino Mega adalah sebuah sistem minimum berbasis mikrokontroler ATmega2560. Arduino Mega 2560 memiliki sejumlah fasilitas untuk berkomunikasi dengan komputer, dengan Arduino lain, atau dengan mikrokontroler lainnya. Arduino Mega 2560 menyediakan 4 hardware komunikasi serial UART TTL (5 Volt). Perangkat lunak Arduino termasuk didalamnya serial monitor memungkinkan data tekstual sederhana dikirim ke dan dari papan Arduino. LED RX dan TX yang tersedia pada papan akan berkedip ketika data sedang dikirim atau diterima melalui *chip* USB-to-serial yang terhubung melalui USB komputer (tetapi tidak untuk komunikasi serial seperti pada pin 0 dan 1). Spesifikasi *board* Arduino berdasarkan datasheet adalah sebagai berikut:

1. Tegangan aktif 5V
2. Tegangan masukan (batas) 6-20V
3. Pin *Input/Output Digital* 54 (6 dapat digunakan sebagai PWM)
4. Pin *Input Analog* 16
5. Arus DC pin I/O 40 mA
6. Arus DC pada 3.3V 50 mA
7. *Flash Memory* 128 KB (ATmega328), 0,5 KB digunakan *bootloader*
8. EEPROM 4 KB (ATmega328)
9. *Clock Speed* 16 MHz

Board ini dilengkapi *jack* DC dengan ukuran tegangan antara 6-20 V. Selain itu, *board* ini juga dilengkapi USB untuk komunikasi serial sehingga dapat dengan mudah dikoneksikan dengan PC. Bentuk fisik *board* Arduino seperti pada Gambar 2.2.



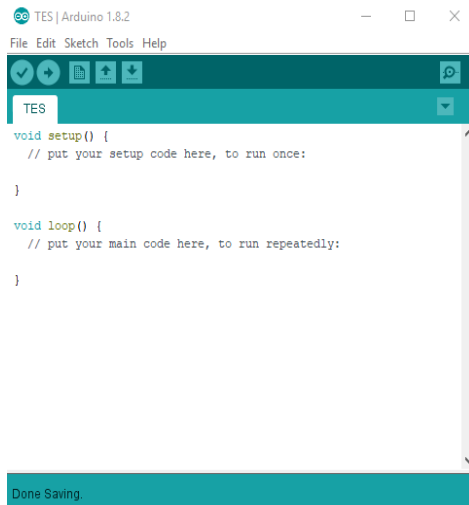
Gambar 2.2 *Board Arduino Mega*

Masing-masing dari 54 digital pin pada Arduino Mega dapat digunakan sebagai *input* atau *output*, menggunakan fungsi `pinMode()`, `digitalWrite()`, dan `digitalRead()`. Arduino Mega beroperasi pada tegangan 5 volt. Setiap pin dapat memberikan atau menerima arus maksimum 40 mA dan memiliki resistor *pull-up* internal (yang terputus secara default) sebesar 20-50 KOhms. Selain itu, beberapa pin memiliki fungsi khusus, antara lain:

1. Serial 0 terdiri dari 0 (RX) dan 1 (TX), Serial 1 terdiri dari 19 (RX) dan 18 (TX), Serial 2 terdiri dari 17 (RX) dan 16 (TX), Serial 3 terdiri dari 15 (RX) dan 14 (TX). Digunakan untuk menerima (RX) dan mengirimkan (TX) data serial TTL. Pin 0 dan 1 juga terhubung ke pin *chip* ATmega16U2 Serial USB-to-TTL.
2. Eksternal Interupsi : Pin 2 (interrupt 0), pin 3 (interrupt 1), pin 18(interrupt 5), pin 19 (interrupt 4), pin 20 (interrupt 3), dan pin 21 (interrupt 2). Pin ini dapat dikonfigurasi untuk memicu sebuah interupsi pada nilai yang rendah, meningkat atau menurun, atau berubah nilai.
3. SPI : Pin 50 (MISO), pin 51 (MOSI), pin 52 (SCK), pin 53 (SS). Pin ini mendukung komunikasi SPI menggunakan perpustakaan SPI. Pin SPI juga terhubung dengan *header* ICSP, yang secara fisik kompatibel dengan Arduino Uno, Arduino Duemilanove, dan Arduino Diecimila.
4. LED : Pin 13. Tersedia secara *built-in* pada papan Arduino ATmega2560. LED terhubung ke pin digital 13. Ketika pin diset bernilai *high*, maka LED menyala (*on*), dan ketika pin diset bernilai *low*, maka LED padam (*off*).
5. TWI : Pin 20 (SDA) dan pin 21 (SCL). Yang mendukung komunikasi TWI menggunakan perpustakaan *wire*.

2.4 Arduino IDE

Board Arduino dapat di program menggunakan *software open source* bawaan Arduino IDE. Arduino IDE adalah sebuah aplikasi *crossplatform* yang berbasis bahasa pemrograman *processing* dan *wiring*. Arduino IDE di desain untuk mempermudah pemrograman dengan adanya kode editor yang dilengkapi dengan *syntax highlighting*, *brace matching*, dan indentasi otomatis untuk kemudahan pembacaan program, serta dapat meng-*compile* dan meng-*upload* program ke *board* dalam satu klik. Jendela Arduino IDE dapat dilihat pada Gambar 2.3.



Gambar 2.3 Jendela Arduino IDE

IDE Arduino adalah *software* yang sangat canggih ditulis dengan menggunakan Java. IDE Arduino terdiri dari:

1. Editor program merupakan sebuah *window* yang memungkinkan pengguna menulis dan mengedit program dalam bahasa *processing*.
2. Compiler merupakan sebuah modul yang mengubah kode program (bahasa *Processing*) menjadi kode biner. Bagaimanapun sebuah *microcontroller* tidak akan bisa memahami bahasa *Processing*. Yang bisa dipahami oleh

microcontroller adalah kode biner. Itulah sebabnya compiler diperlukan dalam hal ini.

3. Uploader merupakan sebuah modul yang memuat kode biner dari komputer ke dalam memory di dalam papan Arduino.

2.5 Sensor Arus SCT013

Transformator arus (CT) adalah sensor yang mengukur arus bolak-balik. Trafo SCT013 merupakan trafo dengan jenis inti split, karena dapat dijepitkan langsung ke kabel bertegangan atau netral yang masuk pada beban yang akan diukur. Seperti trafo lainnya, transformator ini memiliki gulungan primer, inti magnetik, dan gulungan sekunder. Untuk monitoring, gulungan primer digunakan untuk masuknya kabel bertegangan atau netral (hanya salah satu). Dan gulungan sekundernya disambungkan dengan alat pengukur (seperti rangkaian *microcontroller*). Sensor arus SCT013 seperti pada Gambar 2.4.



Gambar 2.4 Sensor Arus SCT013

Arus bolak-balik yang mengalir di primer menghasilkan medan magnet di inti, yang menginduksi arus di belitan sekunder. Arus dalam gulungan sekunder sebanding dengan arus yang mengalir dalam gulungan primer.

2.6 Sensor Tegangan

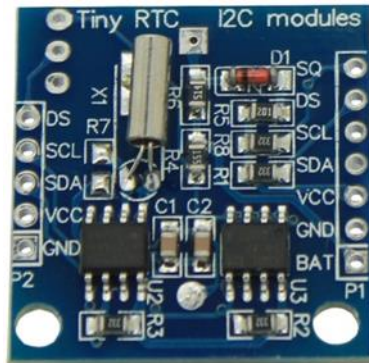
Sensor tegangan merupakan alat yang digunakan untuk mendeteksi besar tegangan yang melalui suatu peralatan listrik. Sensor tegangan menggunakan *transformator* dan rangkaian penyearah. Prinsip kerja dari

sebuah *transformator* adalah ketika kumparan primer dihubungkan dengan sumber tegangan bolak-balik, perubahan arus listrik pada kumparan primer menimbulkan medan magnet yang berubah. Medan magnet yang berubah diperkuat oleh adanya inti besi dan diantarkan inti besi ke kumparan sekunder, sehingga pada ujung-ujung kumparan sekunder akan timbul GGL induksi. Efek ini dinamakan induktansi timbal-balik (*mutual inductance*).

2.7 Real Time Clock DS1307 Modules (RTC)

Real Time Clock (RTC) adalah jenis pewaktu yang bekerja berdasarkan waktu yang sebenarnya atau dengan kata lain berdasarkan waktu yang ada pada jam kita. Meskipun istilah sering mengacu pada perangkat di komputer pribadi, server dan *embedded system*, RTC hadir di hampir semua perangkat elektronik yang perlu untuk menjaga keakuratan waktu. RTC memiliki sumber tenaga alternatif, sehingga mereka dapat terus menjaga waktu sementara sumber utama daya mati atau tidak tersedia. Sumber tenaga alternatif ini biasanya berupa baterai lithium dalam sistem lama, tetapi beberapa sistem yang lebih baru menggunakan *supercapacitor*, karena mereka dapat diisi ulang dan dapat disolder. Sumber daya alternatif juga dapat menyalurkan listrik ke RAM yang didukung baterai. Pada umumnya tenaga alternatif yang digunakan sebesar 3 Volt dari baterai *lithium*.

Kebanyakan RTC menggunakan osilator kristal, tetapi beberapa menggunakan frekuensi saluran listrik. Dalam banyak kasus frekuensi osilator yang digunakan adalah 32,768 kHz. Frekuensi ini sama dengan yang digunakan dalam jam kuarsa dan jam tangan, selain itu frekuensi yang dihasilkan adalah persis 215 siklus per detik, yang merupakan tingkat nyaman untuk digunakan dengan sirkuit biner sederhana. RTC tiny I2C *modules* seperti pada Gambar 2.5.



Gambar 2.5 RTC Tiny I2C Modules

2.8 *Liquid Crystal Display Keypad (LCD Keypad)*

LCD (*Liquid Crystal Display*) berfungsi menampilkan suatu nilai hasil sensor, menampilkan teks, atau menampilkan menu pada aplikasi *microcontroller*. Sumber cahaya di dalam sebuah perangkat LCD adalah lampu neon berwarna putih dibagian belakang susunan kristal cair tadi.

Titik cahaya yang jumlahnya puluhan ribu bahkan jutaan inilah yang membentuk tampilan. Kutub kristal cair yang dilewati arus listrik akan berubah karena pengaruh polarisasi medan magnetik yang timbul dan oleh karenanya akan hanya beberapa warna.

LCD membutuhkan *driver* supaya bisa dikoneksikan dengan sistem minimum dalam suatu *microcontroller*. *Driver* yang disebutkan berisi rangkaian pengaman, pengatur tingkat kecerahan maupun data, serta untuk mempermudah pemasangan di *microcontroller*. *Liquid Crystal Display Keypad (LCD Keypad)* seperti pada Gambar 2.6.



Gambar 2.6 Liquid Crystal Display Keypad (LCD Keypad)

LCD *Keypad* dikembangkan untuk *Arduino Shield*, tujuannya untuk menyediakan antarmuka yang *user-friendly* dan memungkinkan pengguna untuk membuat berbagai pilihan menu dan lainnya. LCD *Keypad* ini terdiri dari 1602 karakter putih *backlight* biru. Terdapat 5 tombol yang terdiri dari *select*, *up*, *right*, *down* dan *left*. Untuk menyimpan pin IO digital, antarmuka keypad hanya menggunakan satu saluran ADC. Berikut adalah konfigurasi pin untuk LCD *Keypad* pada Tabel 2.1.

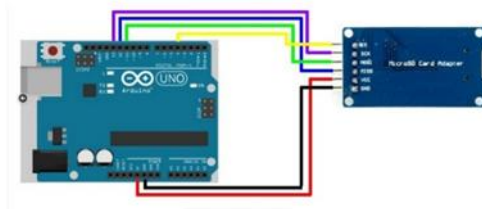
Tabel 2.1 Konfigurasi Pin LCD *Keypad* 16x2

Pin	<i>Function</i>
Analog 0	<i>Button (select, up, right, down and left)</i>
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (<i>Data or Signal Display Selection</i>)
Digital 9	<i>Enable</i>
Digital 10	<i>Backlit Control</i>

2.9 *Module Data Logger*

Sedangkan untuk pencatatan datanya diperlukan memori untuk menyimpan data yang telah terbaca. Memori yang digunakan adalah *micro SD* yang telah terpasang pada modul *micro SD adapter*. Wiring modul *micro SD adapter* dengan *Arduino Mega* seperti pada Gambar 2.7. Berikut adalah sambungan antara *micro SD adapter* dengan *Arduino Mega*:

1. CS (*Chip Select*) – Pin 53 (SS) *Arduino Mega*
2. SCK (*Serial Clock*) – Pin 52 (SCK) *Arduino Mega*
3. MISO (*Serial Data Out*) – Pin 50 (MISO) *Arduino Mega*
4. MOSI (*Serial Data In*) – Pin 51 (MOSI) *Arduino Mega*
5. VCC – Pin 5 Volt *Arduino Mega*
6. GND – Pin *Ground* *Arduino Mega*



Gambar 2.7 Wiring Modul Micro SD Adapter dengan Arduino Mega

2.10 *Miniature Circuit Breaker* (MCB)

Mini Circuit Breaker (MCB) memiliki fungsi sebagai alat pengaman arus lebih. MCB ini memproteksi arus lebih yang disebabkan terjadinya beban lebih dan arus lebih karena adanya hubungan pendek. Prinsip dasar kerjanya yaitu untuk pemutusan hubungan yang disebabkan beban lebih dengan relai arus lebih sesaat menggunakan elektromagnet.

Bila elektromagnet bekerja, maka akan memutus hubungan kontak yang terletak pada pemadam busur dan membuka saklar. MCB untuk rumah seperti pada pengaman lebur diutamakan untuk proteksi hubungan pendek, sehingga pemakaiannya lebih diutamakan untuk mengamankan instalasi atau konduktornya. Arus nominal yang digunakan pada APP dengan mengenal tegangan 230/400 V ialah: 1 A, 2 A, 4 A, 6 A, 10 A, 16 A, 20 A, 25 A, 35 A, dan 50 A disesuaikan dengan tingkat VA konsumen. Adapun kemampuan membuka (breaking capacity) bila terjadi hubung singkat 3 KA dan 6 KA (SPLN 108-1993). MCB yang khusus digunakan oleh PLN mempunyai tombol biru. MCB pada saat sekarang ini paling banyak digunakan untuk instalasi rumah, instalasi industri maupun instalasi gedung bertingkat. *Mini Circuit Breaker* (MCB) seperti pada Gambar 2.8.



Gambar 2.8 *Mini Circuit Breaker* (MCB)

2.11 Ethernet Shield Arduino Module

Ethernet Shield menambahkan kemampuan arduino board agar terhubung ke jaringan komputer. Ethernet shield berbasis cip Ethernet Wiznet W5100. Ethernet library digunakan dalam menulis program agar arduino *board* dapat terhubung ke jaringan dengan menggunakan Arduino Ethernet Shield.

Pada Ethernet Shield terdapat sebuah slot *micro SD*, yang dapat digunakan untuk menyimpan file yang dapat diakses melalui jaringan. *On board micro SD card reader* diakses dengan menggunakan SD library. Arduino *board* berkomunikasi dengan W5100 dan SD *card* menggunakan bus SPI (*Serial Peripheral Interface*). Komunikasi ini diatur oleh library SPI.h dan Ethernet.h Adapun spesifikasi yang dimiliki Ethernet Arduino diperlihatkan seperti pada Tabel 2.2

Tabel 2.2 Spesifikasi Ethernet Arduino

<i>Microcontroller</i>	<i>Atmega328</i>
<i>Operating Voltage</i>	<i>5V</i>
<i>Input Voltage (recommended)</i>	<i>7-12V</i>
<i>Input Voltage (Limit)</i>	<i>6-20V</i>
<i>Digital I/O Pins</i>	<i>14 (of which 4 provide PWM output)</i>
<i>Arduino Pins Reserved</i>	
	<i>10 tp 13 used for SPI</i>
	<i>4 used for SD Card</i>
	<i>2 w5199 interrupt (when bridged)</i>
<i>Analog Input Pins</i>	<i>6</i>
<i>DC Current per I/O Pin</i>	<i>40 mA</i>
<i>DC Current for 3.3V Pin</i>	<i>50 mA</i>
<i>Flash Memory</i>	<i>32 KB of which 0.5 KB used by bootloader</i>
<i>SRAM</i>	<i>2 KB</i>
<i>EEPROM</i>	<i>1 KB</i>
<i>Clock Speed</i>	<i>16 MHz</i>

<i>W5100 TCP/IP Embedded Ethernet Controller</i>	
<i>Power Over Ethernet ready Magnetic Jack</i>	
<i>Micro SD card, with active voltage translator</i>	
<i>Length</i>	<i>68.6 mm</i>
<i>Width</i>	<i>53.3 mm</i>
<i>Weight</i>	<i>28 g</i>

Hardware Ethernet biasanya berupa modul yang dapat dihubungkan ke *port* komputer (paralel, serial, dan USB). Salah satunya produk Ethernet yang dipakai untuk Tugas Akhir ini adalah modul Ethernet dari Arduino yaitu Ethernet Shield Arduino seperti pada Gambar 2.9.

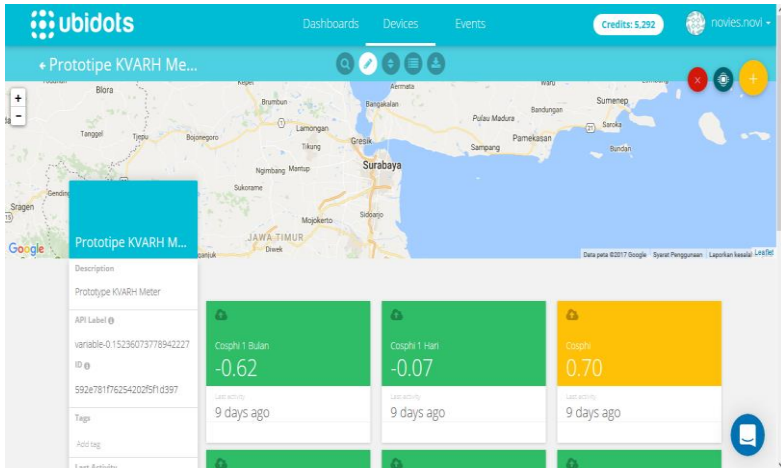


Gambar 2.9 Modul Ethernet Shield

2.12 IOT Platform - Ubidots API

Ubidots menawarkan platform untuk pengembang yang memungkinkan mereka untuk dengan mudah menangkap data sensor dan mengubahnya menjadi informasi yang berguna. Menggunakan platform Ubidots untuk mengirim data ke *cloud* dari perangkat berkemampuan Internet. Selain itu, dapat mengkonfigurasi berupa tindakan dan *alert* berdasarkan data *real time* dan membuka nilai dari data sensor melalui alat visual. Ubidots menawarkan fitur API yang

memungkinkan sensor untuk membaca dan menulis data ke sumber daya yang tersedia: sumber data, variabel, nilai-nilai, peristiwa dan wawasan. API mendukung HTTP dan HTTPS dan Key API diperlukan. Tampilan Ubidots seperti pada Gambar 2.10.



Gambar 2.10 Tampilan Ubidots

BAB III

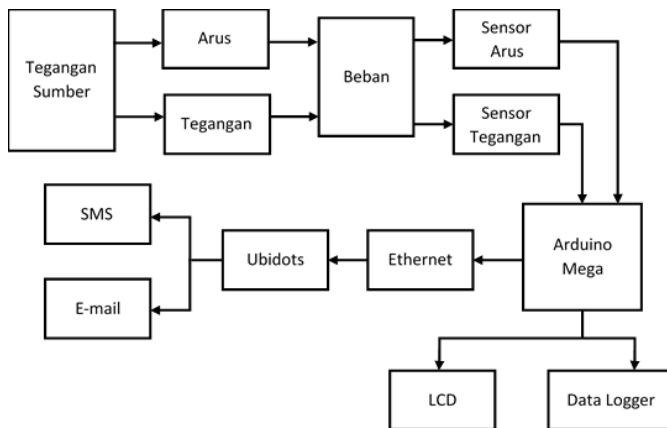
PERANCANGAN DAN PEMBUATAN SISTEM

Pada Bab ini dibahas mengenai perancangan dan pembuatan “Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT)” baik perangkat elektronika (*hardware*) maupun perangkat lunak (*software*) yang meliputi:

1. Perancangan *hardware*: bagian-bagian *hardware* tersebut meliputi sensor arus, sensor tegangan, LCD, RTC, data *logger* dan Ethernet Shield.
2. Perancangan *software*: perancangan perangkat lunak (*software*) yang digunakan yaitu dengan menggunakan program Arduino IDE dan *Internet of Things* (IoT) sebagai media *interface*.

3.1 Gambaran Umum Sistem

Blok fungsional sistem yang di buat pada perancangan Tugas Akhir ini secara keseluruhan dapat dilihat pada Gambar 3.1.



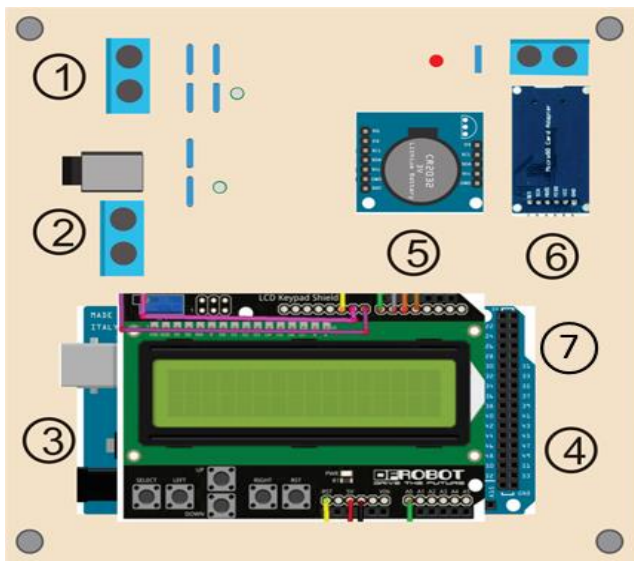
Gambar 3.1 Diagram Fungsional Sistem

Beberapa komponen yang terhubung dengan Arduino Mega adalah sensor arus dan sensor tegangan sebagai *input* serta LCD sebagai *output*. *Output* sensor arus dan sensor tegangan terhubung pada pin analog

Arduino Mega dan dibaca oleh ADC. Data ADC tersebut akan disimpan dalam besaran sesungguhnya pada program Arduino IDE setelah melalui proses linierisasi. Sensor arus dan tegangan terhubung pada pin *interrupt* eksternal Arduino Mega yang akan menjalankan fungsi *timer* sehingga diketahui selisih waktu antara gelombang arus dan tegangan, kemudian didapatkan nilai faktor dayanya. Setelah di temui faktor dayanya ditampilkan pada LCD. Data yang telah terekam pada alat akan di simpan pada data *logger*. Proses pengiriman data menuju Ubidots menggunakan modul Ethernet yang sudah terintegrasi. Dalam kondisi tertentu, Ubidots akan mengirimkan SMS dan E-mail apabila kondisi yang sudah ditentukan terpenuhi.

3.2 Perancangan Mekanik

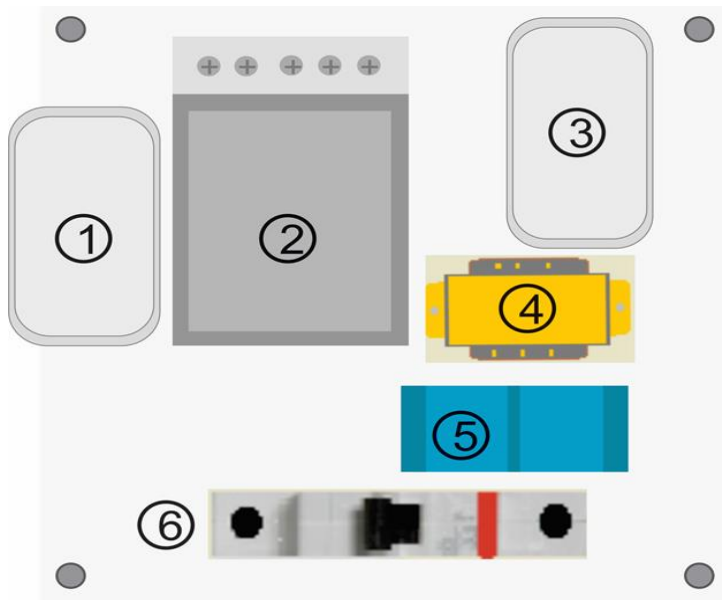
Perancangan mekanik pada prototipe KVARH meter di dalamnya terdapat rangkaian *power supply*, modul Arduino, modul Ethernet, LCD keypad, rangkaian sensor tegangan dan sensor sensor arus. Tata letak perancangan mekanik terdapat pada Gambar 3.2 dan Gambar 3.3 desain luar KVARH meter terdapat pada Gambar 3.4. Untuk foto alat dapat dilihat pada Lampiran C.1.



Gambar 3.2 Perancangan Tata Letak Alat Bagian Atas

Keterangan:

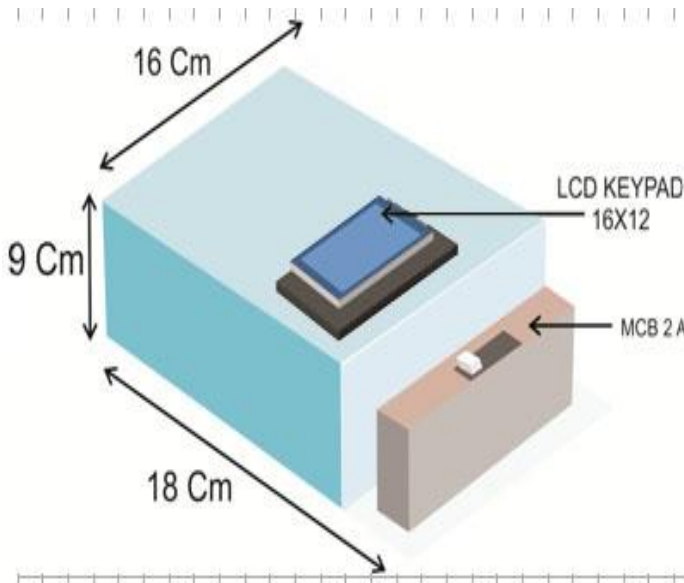
1. Sensor tegangan
2. Sensor Arus
3. Arduino Mega
4. Ethernet Shield
5. RTC
6. Modul micro SD
7. LCD



Gambar 3.3 Perancangan Tata Letak Alat Bagian Bawah

Keterangan:

1. Terminal
2. *Power Supply*
3. Terminal
4. *Tranformator*
5. *Clamper arus*
6. MCB



Gambar 3.4 Desain Bentuk Depan KVARH Meter

Keterangan:

1. LCD 16 x 12 digunakan untuk menampilkan informasi besarnya energi listrik, status konektivitas, informasi waktu, daya aktif, daya reaktif, dan daya semu
2. MCB 2 *Ampere*

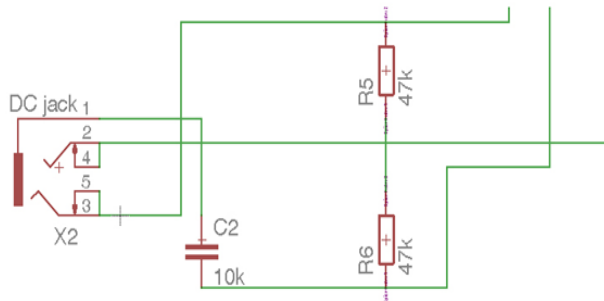
3.3 Perancangan *Hardware*

Tahap pertama perancangan sistem adalah perancangan hardware yang meliputi perancangan sensor arus, perancangan sensor tegangan, perancangan *wiring* keypad LCD, perancangan *wiring* RTC, perancangan *wiring* data logger dan Ethernet Shield.

3.3.1 Perancangan Sensor Arus

Sensor arus merupakan alat yang digunakan untuk mengetahui besarnya arus yang mengalir pada suatu peralatan listrik. Maka pada perancangan ini, digunakan sebuah transformator arus berupa CT YHDC SCT-013-000. YHDC sct-013-000 yang digunakan dengan skala

30 A/1 V, penggunaannya dengan memasang *clamp* salah satu fasanya. Dengan menggunakan *clamp* ini langsung dapat mengukur tegangan AC langsung, dan nantinya akan dibaca dengan *output* analog menggunakan Arduino. Rangkaian Pengkondisional Sensor Arus seperti pada Gambar 3.5.



Gambar 3.5 Rangkaian Pengkondisional Sensor Arus

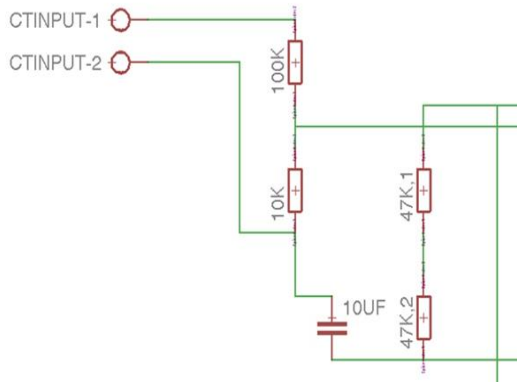
Pada rangkaian sensor arus tersebut menggunakan DC *jack* untuk pembacaan hasil perhitungan analog dari YHDC SCT-013-000. Hasil pembacaan dari sensor arus tersebut dibaca oleh arduino berupa ADC. ADC tersebut dipotong oleh resistor. Tegangan 1 Volt dari output sensor masih memiliki tegangan negatif. Tegangan ini yang nantinya akan dimasukkan pada ADC Arduino. Agar semua data dapat dibaca oleh Arduino, nilai negatif pada tegangan harus diubah mejadi positif. Gelombang yang telah dipotong tersebut, dihilangkan bentuk *noise*-nya dengan menggunakan kapasitor 10uF tersebut. Dengan begitu Arduino dapat membaca nominal ADC dengan baik dan jelas.

3.3.2 Perancangan Sensor Tegangan

Tugas akhir ini menggunakan sensor tegangan dari *transformer step down* 220 Volt/12 Volt. Trafo *stepdown* digunakan untuk menurunkan tegangan 220 Volt agar nantinya dapat dibaca oleh ADC pada arduino yang membutuhkan tegangan 0-5 Volt.

Agar hasil dari sensor tegangan dapat dibaca oleh ADC Arduino maka dilakukan pembagian tegangan agar didapat nilai tegangan yang lebih kecil. Tegangan yang akan masuk pada ADC masih berupa gelombang sinus yang memiliki nilai negatif. Agar semua data

dapat dibaca oleh arduino, nilai negatif pada tegangan harus diubah mejadi positif. Berikut rangkaian untuk mengkonversi tegangan 12 Volt agar dapat dibaca arduino. Perancangan Sensor Tegangan seperti pada Gambar 3.6.



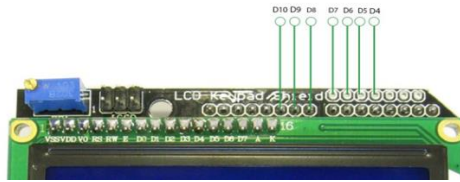
Gambar 3.6 Perancangan Sensor Tegangan

3.3.3 Perancangan *Wiring keypad* LCD

Liquid Crystal Display atau LCD digunakan untuk mempermudah *user* untuk melihat kondisi terkini dari relai. LCD yang digunakan adalah LCD *Keypad* 16 x 2. Perancangan *wiring* LCD keypad pada Arduino Mega seperti pada Gambar 3.7.

Berikut adalah pengkabelan yang harus dilakukan untuk menghubungkan LCD *Keypad* dengan Arduino Mega 2560:

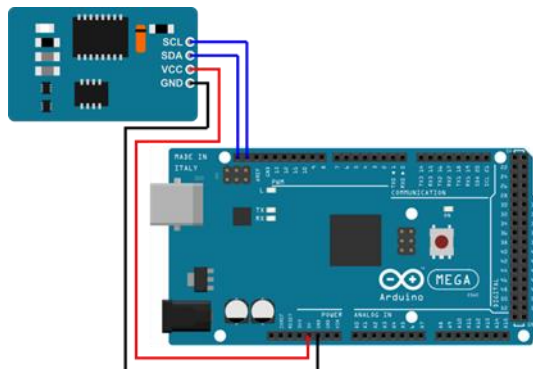
1. Pin A0 dengan pin 5 tombol LCD *Keypad*
2. Pin Digital 4 dihubungkan dengan pin DB 4
3. Pin Digital 5 dihubungkan dengan DB5
4. Pin Digital 6 dihubungkan dengan DB 6
5. Pin Digital 7 dihubungkan dengan DB7
6. Pin Digital 8 dihubungkan dengan pin RS
7. Pin Digital 9 dihubungkan dengan pin *Enable*
8. Pin Digital 10 dihubungkan dengan pin *Backlight Control*



Gambar 3.7 Perancangan *Wiring* LCD Keypad pada Arduino Mega

3.3.4 Perancangan *Wiring* RTC

Modul RTC menggunakan komunikasi I2C untuk bertukar data dengan Arduino Mega. Modul ini memiliki 4 pin yang harus terhubung dengan Arduino Mega. *Wiring* hubungan antara RTC dengan Arduino Mega dapat dilihat pada Gambar 3.8.



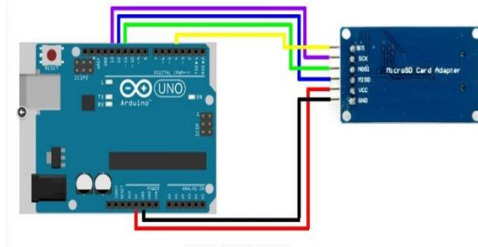
Gambar 3.8 *Wiring* RTC pada Arduino Mega

1. Pin VCC terhubung dengan referensi tegangan 5 volt Arduino Mega
2. Pin GND terhubung dengan *ground* Arduino Mega
3. Pin SDA terhubung dengan pin SDA Arduino Mega
4. Pin SCL terhubung dengan pin SCL Arduino Mega

3.3.5 Perancangan Wiring Data Logger

Untuk pencatatan datanya diperlukan memori untuk menyimpan data yang telah terbaca. Memori yang digunakan adalah *micro SD* yang telah terpasang pada modul *micro SD adapter*. Wiring data logger seperti pada Gambar 3.9. Berikut adalah sambungan antara *micro SD adapter* dengan Arduino Mega:

1. CS (*Chip Select*) – Pin 53 (SS) Arduino Mega
2. SCK (*Serial Clock*) – Pin 52 (SCK) Arduino Mega
3. MISO (*Serial Data Out*) – Pin 50 (MISO) Arduino Mega
4. MOSI (*Serial Data In*) – Pin 51 (MOSI) Arduino Mega
5. VCC – Pin 5 Volt Arduino Mega
6. GND – Pin *Ground* Arduino Mega



Gambar 3.9 Wiring Data Logger

3.3.6 Ethernet Shield Arduino

Pada perancangan perangkat keras prototipe KVARH meter terdapat mikrokontroler Arduino yang terhubung dengan modul Ethernet, Arduino Mega 2560 digunakan sebagai *controller* dari sistem ini, sedangkan modul Ethernet dibutuhkan sebagai media komunikasi ke jaringan internet yang nantinya akan terhubung ke router melalui kabel RJ45.

Pin-pin yang digunakan oleh modul Ethernet telah dibuat untuk menyesuaikan dengan mikrokontroller Arduino Mega. Untuk pemasangan dari modul Ethernet ke Arduino mega hanya ditancapkan saja mulai dari pin 0 (RX0) hingga AREF dan pin A5 hingga RESET. Berikut gambar dari pemasangan Arduino Mega 2560 dan modul Ethernet. Skematik Ethernet seperti pada Gambar 3.10.



Gambar 3.10 Skematik Ethernet

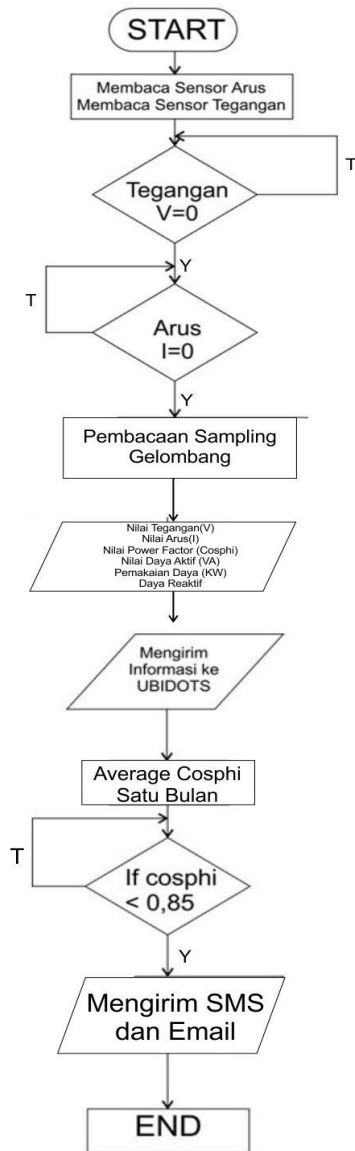
3.4 Perancangan Software

Perangkat *software* diperlukan dalam perancangan sistem sebagai *interface* antara *plant* dan komputer. *Software* yang digunakan dalam pengerjaan Tugas Akhir ini antara lain Arduino IDE. Sedangkan Ubidots digunakan sebagai *interface*.

3.4.1 Perancangan Sistem pada Arduino Mega

Pada Arduino Mega meliputi proses pengambilan data sensor, pengolahan data, dan proses kontrol peralatan. Flowchart perancangan sistem Prototipe KVARH Meter Terintegrasi dengan Web terdapat pada Gambar 3.11.

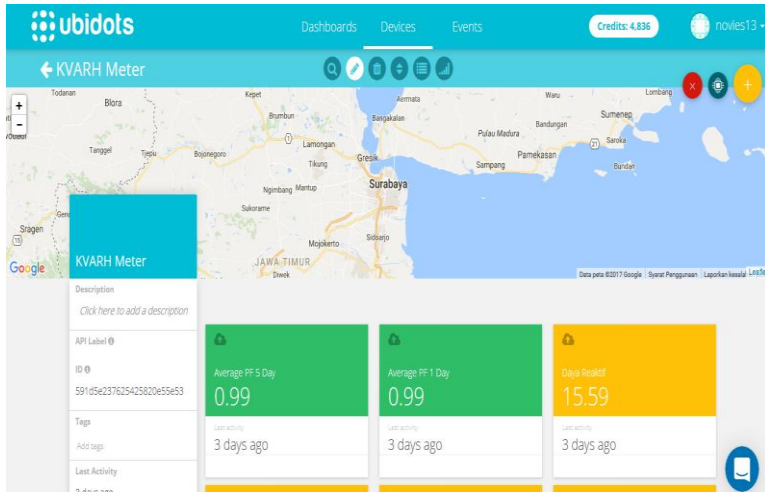
Pada pemrograman Arduino ini membaca data ADC pada sensor arus dan tegangan. Hasil dari pembacaan sensor arus dan tegangan kemudian dilakukan pembacaan yang di program untuk membaca $\cos\phi$, KWH, dan KVARH. Data yang didapat akan dikirimkan ke Ubidots. Pada Ubidots diatur untuk batas minimal faktor daya. Batas minimal yang telah ditentukan oleh PLN yaitu 0,85. Batas minimal ini merupakan batas penggunaan faktor daya pada beban induktif. Apabila sudah memenuhi kondisi yang sudah ditentukan, maka Ubidots akan mengirim informasi melalui SMS dan E-mail.



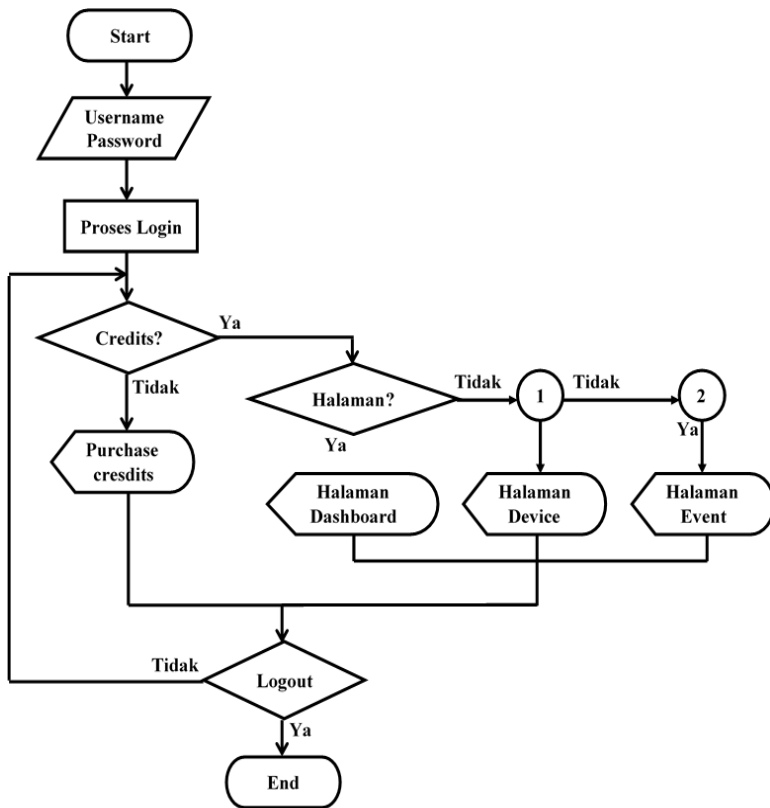
Gambar 3.11 Flowchart Pemrograman Software Arduino IDE

3.4.2 Perancangan *Internet of Things (IoT)*

Perancangan *Internet of Things (IoT)* digunakan sebagai media *interface* alat dengan operator. *Platform IoT* yang digunakan adalah Ubidots. Berikut ini merupakan tampilan ubidots pada Gambar 3.12. Sedangkan flowchart alur tampilan Ubidots pada Gambar 3.13. Untuk tampilan lebih lengkapnya dapat dilihat pada Lampiran C.2.



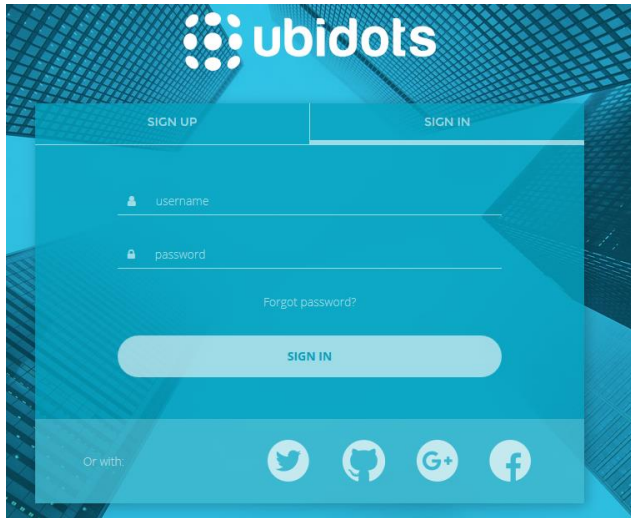
Gambar 3.12 Tampilan Ubidots



Gambar 3.13 Flowchart Alur Tampilan Ubidots

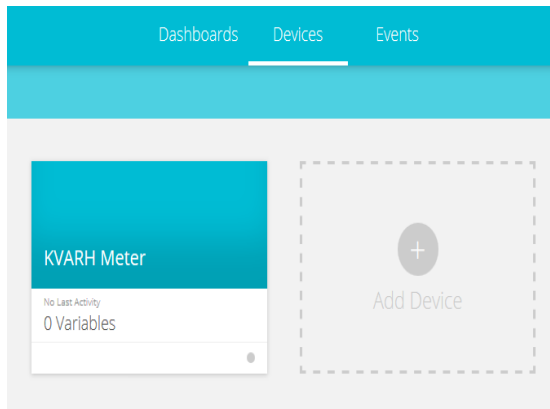
Berikut adalah langkah-langkah untuk koneksi sistem dengan Ubidots:

Buka situs Ubidots pada alamat www.ubidots.com maka akan terbuka halaman seperti pada Gambar 3.14. Klik sign in pada kolom atas dan masukkan username serta password. Tekan tombol sign in berwarna putih atau tekan enter pada keyboard setelah memasukkan password.



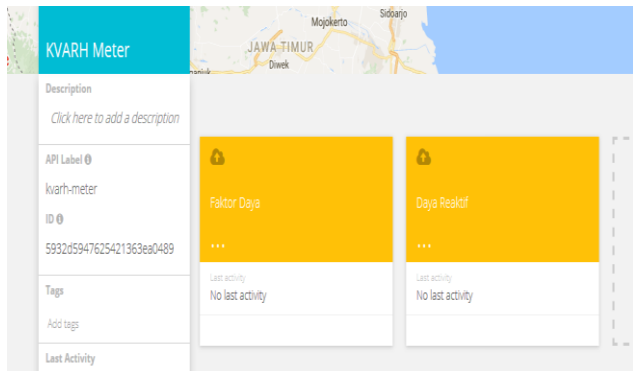
Gambar 3.14 Tampilan Sign Up/Log In Ubidots

Setelah melakukan proses log in pilih tab device seperti ditunjukkan pada Gambar 3.15. Kemudian klik *add device* untuk menambah perangkat baru kedalam *ubidots*. Ganti nama perangkat sesuai yang diinginkan. Klik *device* yang baru saja dibuat untuk membuat variabel baru.



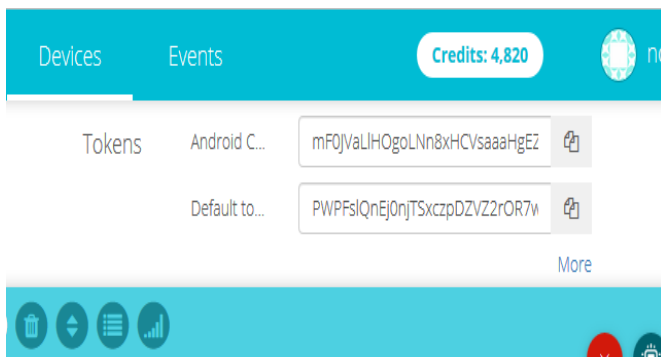
Gambar 3.15 Tampilan *Devices* Ubidots

Setelah *device* diklik maka akan muncul tampilan seperti pada Gambar 3.16. Klik add variable kemudian pilih default pada toolbar yang muncul. Ganti nama variable sesuai yang diinginkan. Untuk menambahkan variable lain cukup ulangi langkah ini kembali pada ikon Add Variable yang muncul.



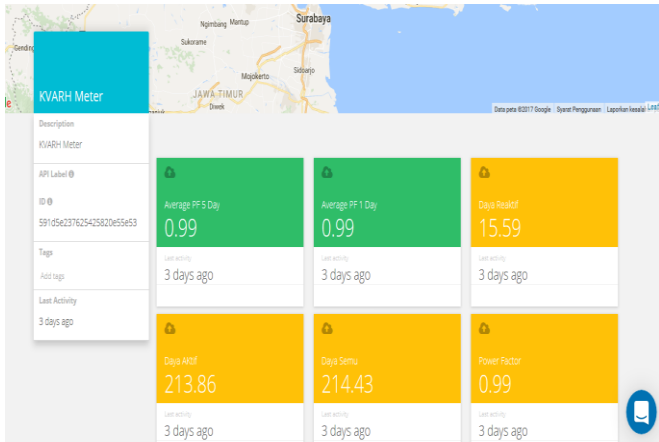
Gambar 3.16 Tampilan *Variable* Ubidots

Klik username pada pojok kanan atas. Pilih API Credentials pada toolbar yang muncul untuk menampilkan token seperti pada Gambar 3.17. Copy token ke program arduino IDE. Selanjutnya pada tampilan variable, klik ikon info untuk menampilkan variable ID. Copy variable ID ke program arduino IDE.



Gambar 3.17 Tampilan *Token* Ubidots

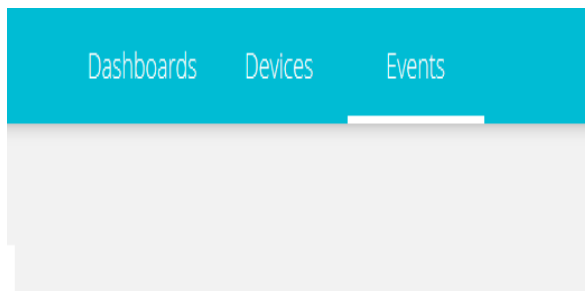
Pastikan jaringan dalam kondisi baik. Upload program Arduino Ide, maka data akan tampil pada halaman ubidots seperti pada Gambar 3.18.



Gambar 3.18 Tampilan *Device* Ubidots

Agar ubidots dapat mengirimkan notifikasi kepada pengguna maka dilakukan konfigurasi sistem notifikasi. Berikut adalah langkah-langkah untuk melakukan konfigurasi sistem notifikasi dengan Ubidots:

Pada *menu bar*, klik bagian events sehingga muncul tampilan seperti pada Gambar 3.19. Klik ikon add event untuk menambahkan event baru.



Gambar 3.19 Tampilan Menu Bar Ubidots

Pilih device dan variable yang akan diatur seperti pada Gambar 3.20. Tentukan *logic value* (IF THAN Logic) sesuai yang diinginkan. Notifikasi akan dikirimkan sesuai *logic value* yang ditentukan. Klik continue untuk melanjutkan.

The screenshot shows the 'if' configuration screen in Ubidots. At the top, there is a cloud icon with an upward arrow, a dropdown menu labeled 'Value', and the text 'Power Factor'. Below this, there are five comparison operators: 'less' (represented by a blue less-than sign), 'greater' (represented by a blue greater-than sign), 'less or equal' (represented by a blue less-than-or-equal sign), 'greater or equal' (represented by a blue greater-than-or-equal sign), and 'equal' (represented by a blue equals sign). The 'less' operator is currently selected. Below the operators, the word 'than' is displayed. Underneath 'than' is a text input field containing the value '0.85'. At the bottom of the screen, there is a blue button labeled 'Continue'.

Gambar 3.20 Tampilan *IF THAN Logic* Ubidots

Untuk mengirimkan pesan melalui SMS seperti pada Gambar 3.21. Pilih media Send SMS. Masukkan nomor tujuan yang akan dikirimkan notifikasi pada kolom phone number, tambahkan nomor jika lebih dari satu tujuan. Tulis pesan yang akan dikirimkan pada kolom message. Klik finish untuk menyelesaikan konfigurasi.

Send e-mail Send SMS Send Telegram WebHook Set a variable

+ Add number

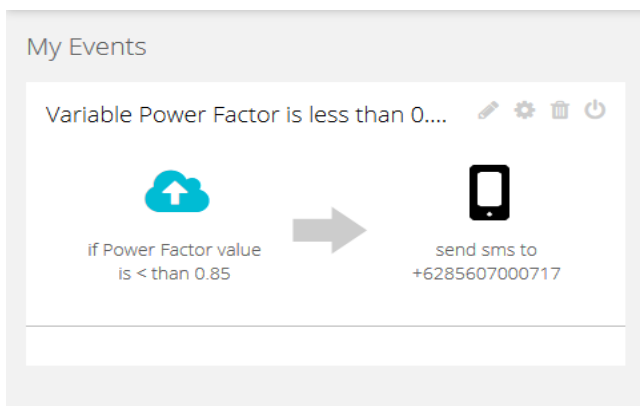
Phone number Indonesia +62 85607000717

Message Hi novies13, variable Power Factor is less than 0.85

Finish

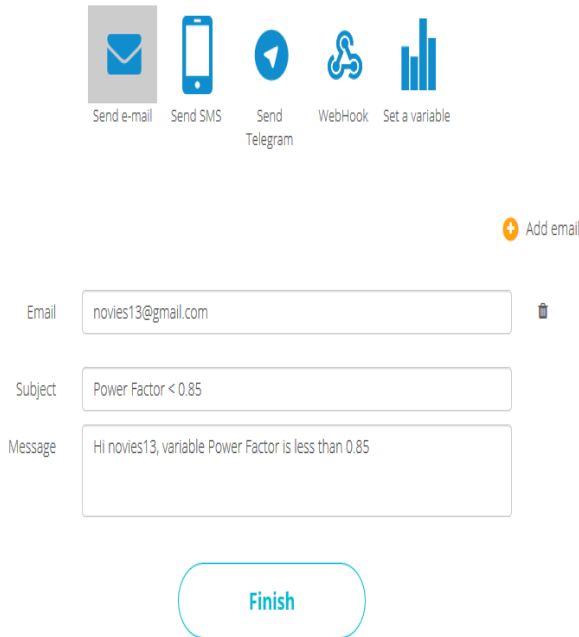
Gambar 3.21 Tampilan Pengiriman SMS Ubidots

Setelah menekan tombol finish muncul tampilan seperti seperti pada Gambar 3.22. Pesan akan dikirimkan apabila memenuhi nilai logic yang telah ditentukan.



Gambar 3.22 Tampilan *Event* Pengiriman SMS Ubidots

Untuk mengirimkan pesan melalui e-mail pilih media Send e-mail seperti pada Gambar 3.23. Masukkan alamat E-mail yang dituju pada kolom yang telah ditentukan. Tambahkan alamat e-mail jika lebih dari satu tujuan. Tulis subject dan message yang akan dikirim. Klik finish untuk menyelesaikan.



Send e-mail Send SMS Send Telegram WebHook Set a variable

+ Add email

Email: novies13@gmail.com

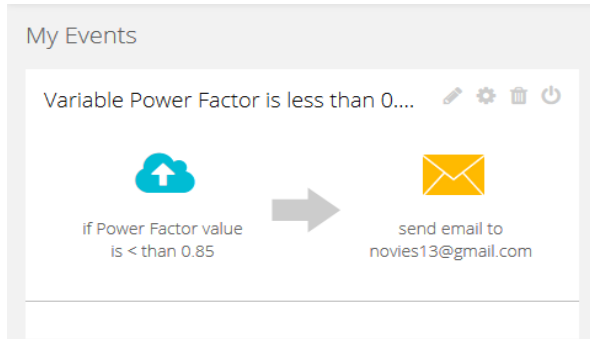
Subject: Power Factor < 0.85

Message: Hi novies13, variable Power Factor is less than 0.85

Finish

Gambar 3.23 Tampilan Pengiriman E-mail Ubidots

Setelah menekan tombol finish muncul tampilan seperti seperti pada Gambar 3.24. Pesan akan dikirimkan apabila memenuhi nilai logic yang telah ditentukan.



Gambar 3.24 Tampilan *Event* Pengiriman E-mail Ubidots

-----Halaman ini sengaja dikosongkan-----

BAB 4

PENGUJIAN DAN ANALISA

Pada Tugas Akhir ini untuk mengetahui kinerja dari peralatan dan pembuatan sistem yang telah dirancang dan direncanakan, maka diperlukan pengujian dan analisa dari setiap komponen pendukung yang dibuat agar sistem dapat berjalan dengan baik sesuai dengan yang diharapkan. Pada bab ini akan dibahas tentang pengujian dan analisa pada *hardware* dan *software* yang telah dibuat.

4.1 Pengujian *Ethernet Shield*

Pada pengujian *Ethernet Shield* untuk melihat kestabilan dari jaringan LAN atau jaringan internet. Pengujian dilakukan dengan cara pengujian ping pada jaringan LAN. Ping adalah waktu reaksi koneksi seberapa cepat computer mendapat jawaban setelah mengirim permintaan pada jaringan computer lain atau jaringan internet. Cara pengujian ping pada jaringan dapat dilakukan melalui command prompt. Berikut langkah-langkah untuk pengujian:

1. Buka command prompt pada windows.
2. Untuk mengetahui besar ping harus mengetahui dahulu nomer IP dari jaringan yang digunakan.
3. IP dapat dicari dengan cara masuk control panel, kemudian pilih network and internet, kemudian pilih network sharing center, kemudian klik kiri pada jaringan yang sudah terkoneksi kemudian pilih details. Nomer IP dapat dilihat pada table tersebut.
4. Setelah mendapatkan nomer IP langkah selanjutnya adalah memberi perintah pada command prompt ketik “ping no IP” kemudian tekan enter.
5. Informasi tentang ping akan tampil pada layar command prompt. Hasil test ping dapat dilihat pada Gambar 4.1 dan Gmabar 4.2.

Pengujian dilakukan sebanyak empat kali dengan jarak uji antara router dan ethernet yang berbeda yaitu dengan jarak 1 meter dan 20 meter. Data pengujian dapat dilihat pada Tabel 4.1 dan Tabel 4.2.

Tabel 4.1 Pengujian Ping pada Jarak 1 Meter

No.	Byte	Time (ms)	TTL
1	32	1	128
2	32	1	128
3	32	1	128
4	32	1	128

```
C:\Users\oxipu>ping 192.168.1.17

Pinging 192.168.1.17 with 32 bytes of data:
Reply from 192.168.1.17: bytes=32 time=1ms TTL=128
Reply from 192.168.1.17: bytes=32 time=1ms TTL=128
Reply from 192.168.1.17: bytes=32 time=1ms TTL=128
Reply from 192.168.1.17: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.17:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
```

Gambar 4.1 Pengujian Besar Nilai Ping pada Jaringan Internet Jarak 1 meter

Tabel 4.2 Pengujian Ping pada Jarak 20 Meter

No.	Byte	Time (ms)	TTL
1	32	173	128
2	32	16	128
3	32	71	128
4	32	11	128

```
C:\Users\oxipu>ping 192.168.1.17

Pinging 192.168.1.17 with 32 bytes of data:
Reply from 192.168.1.17: bytes=32 time=173ms TTL=128
Reply from 192.168.1.17: bytes=32 time=16ms TTL=128
Reply from 192.168.1.17: bytes=32 time=71ms TTL=128
Reply from 192.168.1.17: bytes=32 time=11ms TTL=128

Ping statistics for 192.168.1.17:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 11ms, Maximum = 173ms, Average = 67ms
```

Gambar 4.2 Pengujian Besar Nilai Ping pada Jaringan Internet Jarak 20 Meter

Pada Tabel 4.1 dan Tabel 4.2 dapat dilihat bahwa pada pengukuran dengan jarak yang dekat yaitu 1 meter data pengujian ping tampak sangat bagus dengan transfer time 1 ms. Sedangkan pada pengujian ping dengan jarak yang lebih jauh yaitu 20 meter didapatkan data yang kurang bagus sebab transfer time terjadi ketidakstabilan yaitu pada pengiriman data pertama membutuhkan waktu 173 ms, data kedua membutuhkan waktu 16 ms, data ketiga membutuhkan waktu pengiriman sebesar 71 ms dan data keempat membutuhkan waktu pengiriman sebesar 11 ms. Namun dari pengujian ping diatas tidak terdapat loss data. Sehingga jaringan masih dalam kondisi baik.

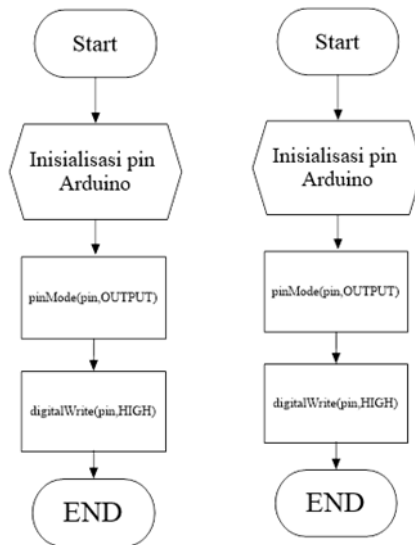
Selain pengujian tes ping, juga dilakukan pengujian pada pin yang digunakan pada sistem alat ini yaitu pin A0 s/d A5, D2 s/d D13 milik Arduino Mega 2560 dengan cara mengatur keluarannya pada logika 1 dan 0 dengan mengunggah program ke *board* Arduino. Pada kondisi 1 memberikan tegangan pada pin Arduino berlogika 1 atau sebesar 5 Volt sedangkan pada kondisi 0 memberikan tegangan keluaran pada pin Arduino berlogika 0 atau 0 Volt. Hasil pengujian dapat dilihat pada Tabel 4.3.

Tabel 4.3 Pengujian Pin Ethernet

No	Pin	V output	
		Kondisi 0	Kondisi 1
1	Analog A0	0,4	4,73
2	Analog A1	1,3	4,76
3	Analog A2	0,4	4,74
4	Analog A3	0,9	4,74
5	Analog A4	0,3	4,76
6	Analog A5	0,4	4,77
12	Digital 2	4,4	4,73
13	Digital 3	4,2	4,76
14	Digital 4	5,2	4,72
15	Digital 5	5,8	4,76
16	Digital 6	0,2	4,79
17	Digital 7	0,2	4,76
18	Digital 8	0,1	4,76
19	Digital 9	0,2	4,76
20	Digital 10	0,1	4,76
21	Digital 11	0,1	4,76
22	Digital 12	0,6	4,74
23	Digital 13	0,2	4,63

4.2 Pengujian Arduino Mega 2560

Pengujian ini dilakukan terhadap *board* Arduino Mega 2560. Pengujian dilakukan pada pin yang digunakan pada sistem alat ini yaitu pin A0 s/d A7, D0 s/d D21 milik Arduino Mega 2560 dengan cara mengatur keluarannya pada logika 1 dan 0 dengan mengunggah program ke *board* Arduino. Flowchart program arduino dapat dilihat pada Gambar 4.3. untuk Gambar 4.3 (a) merupakan program untuk memberikan tegangan pada pin Arduino berlogika 1 atau sebesar 5 Volt sedangkan pada Gambar 4.3 (b) merupakan program untuk memberikan tegangan keluaran pada pin Arduino berlogika 0 atau 0 Volt. Hasil pengujian dapat dilihat pada Tabel 4.4. dan Tabel 4.5.



Gambar 4.3 Flowchart Pemrograman Pengujian pada Pin Arduino (a) Kondisi Logika 1 (b) Kondisi Logika 0

Tabel 4.4 Pengujian Pin Arduino Kondisi Logika 1

No	Pin dan Status	V Output
1	<code>digitalWrite(A0,HIGH);</code>	5,1
2	<code>digitalWrite(A1,HIGH);</code>	5,1
3	<code>digitalWrite(A2,HIGH);</code>	5,1
4	<code>digitalWrite(A3,HIGH);</code>	5,1
5	<code>digitalWrite(A4,HIGH);</code>	5,2
6	<code>digitalWrite(A5,HIGH);</code>	5,1
7	<code>digitalWrite(A6,HIGH);</code>	5,1
8	<code>digitalWrite(A7,HIGH);</code>	5,2
9	<code>digitalWrite(0,HIGH);</code>	5,2
10	<code>digitalWrite(1,HIGH);</code>	5,1
11	<code>digitalWrite(2,HIGH);</code>	5,1
12	<code>digitalWrite(3,HIGH);</code>	5,1
13	<code>digitalWrite(4,HIGH);</code>	5,1

14	digitalWrite(5,HIGH);	5,1
15	digitalWrite(6,HIGH);	5,1
16	digitalWrite(7,HIGH);	5,2
17	digitalWrite(8,HIGH);	5,1
18	digitalWrite(9,HIGH);	5,1
19	digitalWrite(10,HIGH);	5,2
20	digitalWrite(11,HIGH);	5,1
21	digitalWrite(12,HIGH);	5,2
22	digitalWrite(13,HIGH);	5,1

Tabel 4.5 Pengujian Pin Arduino Kondisi Logika 0

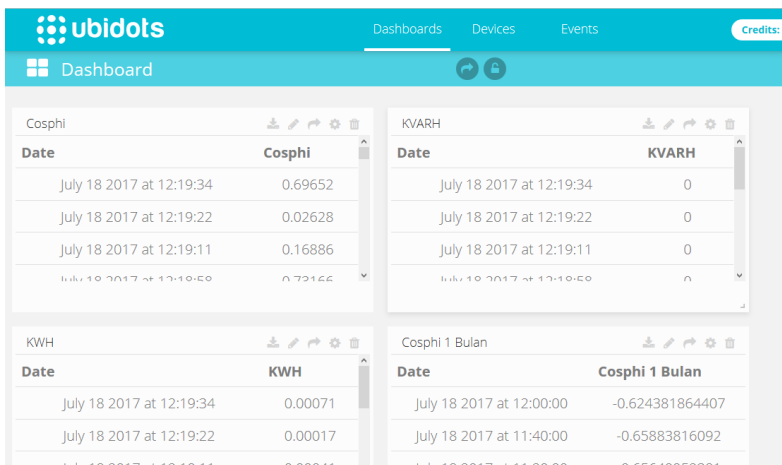
No	Pin dan Status	V Output
1	digitalWrite(A0,HIGH);	0.7 mV
2	digitalWrite(A1,HIGH);	0.7 mV
3	digitalWrite(A2,HIGH);	0.7 mV
4	digitalWrite(A3,HIGH);	0.5 mV
5	digitalWrite(A4,HIGH);	0.7 mV
6	digitalWrite(A5,HIGH);	0.21 mV
7	digitalWrite(A6,HIGH);	0.52 mV
8	digitalWrite(A7,HIGH);	0.21 mV
9	digitalWrite(0,HIGH);	0.46 mV
10	digitalWrite(1,HIGH);	0.3 mV
11	digitalWrite(2,HIGH);	0.3 mV
12	digitalWrite(3,HIGH);	0.23 mV
13	digitalWrite(4,HIGH);	0.45 mV
14	digitalWrite(5,HIGH);	0.23 mV
15	digitalWrite(6,HIGH);	0.23 mV
16	digitalWrite(7,HIGH);	0.21 mV
17	digitalWrite(8,HIGH);	0.4 mV
18	digitalWrite(9,HIGH);	0.3 mV
19	digitalWrite(10,HIGH);	0.52 mV
20	digitalWrite(11,HIGH);	0.4 mv
21	digitalWrite(12,HIGH);	0.3 mV
22	digitalWrite(13,HIGH);	0.4 mV

4.3 Pengujian *Internet of Things (IoT)*

Pada tahap pengujian *Internet of Things (IoT)* dibagi menjadi dua bagian. Bagian pertama adalah pengujian keberhasilan pengiriman data ke Ubidots. Bagian kedua adalah pengujian keberhasilan notifikasi melalui media SMS dan *e-mail*.

4.3.1 Pengujian Pengiriman Data ke Ubidots

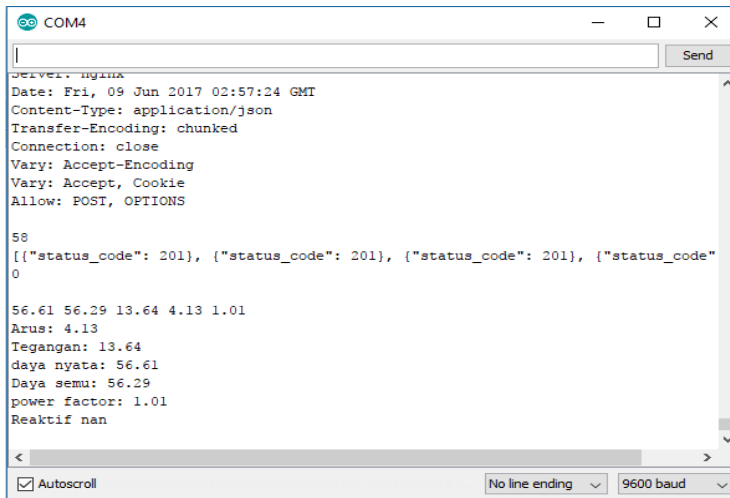
Tujuan pengujian pengiriman data ke Ubidots adalah untuk mengetahui apakah data pada *device* yang telah dibuat telah terintegrasi dengan baik ke Ubidots. Pengujian dilakukan dengan cara membandingkan data yang terbaca pada serial monitor dan data yang terunggah pada Ubidots. Tampilan dashboard pada Ubidots ditunjukkan pada Gambar 4.4. Tampilan data pada serial monitor ditunjukkan pada Gambar 4.5.



The screenshot shows the Ubidots dashboard with a teal header. The main content area displays four data tables arranged in a 2x2 grid. Each table has a title, a 'Date' column, and a value column. The tables are for devices 'Cosphi', 'KVARH', 'KWH', and 'Cosphi 1 Bulan'. The data is as follows:

Device	Date	Value
Cosphi	July 18 2017 at 12:19:34	0.69652
	July 18 2017 at 12:19:22	0.02628
	July 18 2017 at 12:19:11	0.16886
	July 18 2017 at 12:19:00	0.72166
KVARH	July 18 2017 at 12:19:34	0
	July 18 2017 at 12:19:22	0
	July 18 2017 at 12:19:11	0
	July 18 2017 at 12:19:00	0
KWH	July 18 2017 at 12:19:34	0.00071
	July 18 2017 at 12:19:22	0.00017
	July 18 2017 at 12:19:11	0.00041
	July 18 2017 at 12:19:00	0.00041
Cosphi 1 Bulan	July 18 2017 at 12:00:00	-0.624381864407
	July 18 2017 at 11:40:00	-0.65883816092
	July 18 2017 at 11:20:00	0.65640052281
	July 18 2017 at 11:00:00	0.65640052281

Gambar 4.4 Tampilan Dashboard pada Ubidots



Gambar 4.5 Tampilan Data pada Serial Monitor

Perbandingan data pada serial monitor dan Ubidots ditunjukkan pada Tabel 4.6 dan Tabel 4.7. Dari tabel tersebut dapat dilihat bahwa terdapat perbedaan antara data pada serial monitor dan data pada Ubidots. Sebagai contoh data pada serial monitor mencatat terdapat 10 data selama kurang lebih 1 menit, sedangkan pada Ubidots terdapat 5 data selama 1 menit.

Tabel 4.6 Data yang Terkirim pada Ubidots

No.	Tanggal / Waktu	Faktor Daya
1	16 Juni 2017 / 14.44.24	0,75323
2	16 Juni 2017 / 14.44.20	0,75127
3	16 Juni 2017 / 14.44.17	0,75479
4	16 Juni 2017 / 14.44.14	0,75283
5	16 Juni 2017 / 14.44.11	0,74571
6	16 Juni 2017 / 14.43.47	0,97852
7	16 Juni 2017 / 14.43.44	0,97854
8	16 Juni 2017 / 14.43.41	0,9808
9	16 Juni 2017 / 14.43.39	0,98018
10	16 Juni 2017 / 14.43.36	0,98055

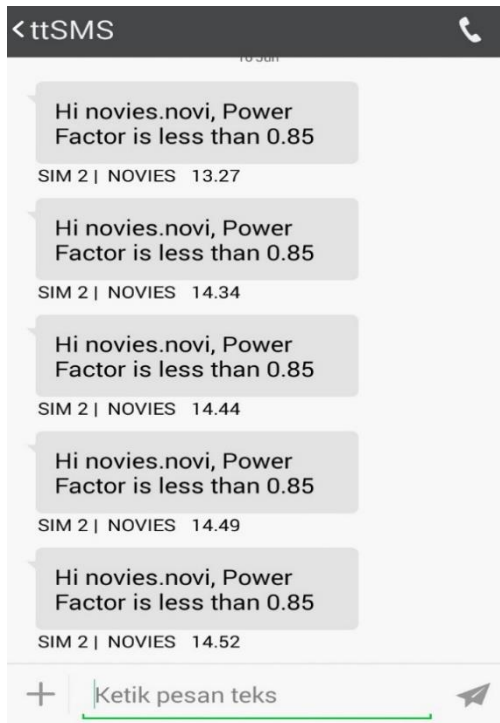
Tabel 4.7 Data pada Serial Monitor

No.	Tanggal / Waktu	Faktor Daya
1	16 Juni 2017 / 14.44.24	0.75
2	16 Juni 2017 / 14.44.20	0.75
3	16 Juni 2017 / 14.44.17	0.75
4	16 Juni 2017 / 14.44.14	0.75
5	16 Juni 2017 / 14.44.11	0.74
6	16 Juni 2017 / 14.43.47	0.97
7	16 Juni 2017 / 14.43.44	0.97
8	16 Juni 2017 / 14.43.41	0.98
9	16 Juni 2017 / 14.43.39	0.98
10	16 Juni 2017 / 14.43.36	0.98

Dapat diambil kesimpulan bahwa data yang dikirim pada Ubidots tingkat ketelitian bilangan desimalnya 4-5 angka di belakang koma. Sedangkan pada serial monitor angka ketelitian bilangan desimalnya hanya 2 angka di belakang koma.

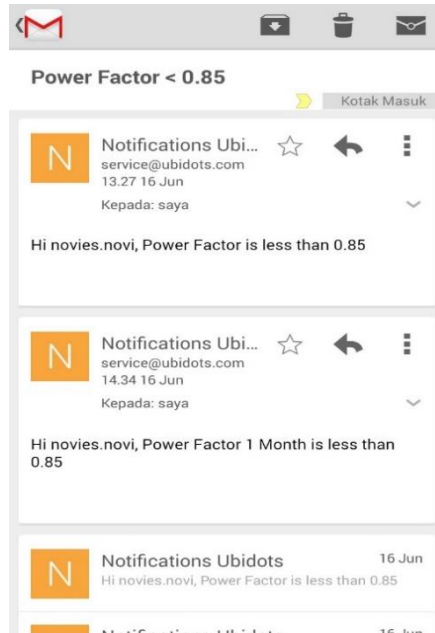
4.3.2 Pengujian Pengiriman Notifikasi

Ubidots akan mengirimkan notifikasi apabila memenuhi kondisi yang sudah ditentukan. Dalam Tugas Akhir ini ubidots akan mengirim notifikasi apabila faktor daya kurang dari 0,85. Pengiriman Notifikasi melalui media SMS dan e-mail yang terdapat pada sistem Ubidots. Berikut adalah contoh pengiriman notifikasi yang ditunjukkan pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Tampilan Notifikasi Melalui SMS

Dapat dilihat pada Gambar 4.6 bahwa ubidots telah berhasil mengirimkan notifikasi melalui SMS. SMS terkirim ketika sudah memenuhi kondisi yang sudah ditentukan. Pada Gambar 4.7 bahwa ubidots telah berhasil mengirimkan notifikasi melalui e-mail. E-mail terkirim ketika sudah memenuhi kondisi yang sudah ditentukan. Pada pengiriman notifikasi, Ubidots akan melakukan pengiriman apabila sudah memenuhi kondisi yang sudah ditentukan. Dalam hal ini keberhasilan pengiriman notifikasi 100%.



Gambar 4.7 Tampilan Notifikasi Melalui e-mail

4.4 Analisa Relevansi

Prototipe KVARH meter jika diimplementasikan pada kehidupan sehari-hari di industri, perlu banyak yang disesuaikan kembali. Pada prototipe ini hanya menggunakan permodelan satu fasa, dengan sumber tegangan 220 Volt dan menggunakan batasan arus dengan MCB sebesar 2A. Pada realitanya bahwa pelanggan industri menggunakan sumber 3 fasa. Dan permodelan perhitungan KVARH dengan satuan waktu berupa jam lebih dipersingkat menjadi menit agar dapat melihat hasil lebih cepat tanpa mengurangi keakuratan perhitungan. Kemudian untuk sistem rekapitulasinya pun menggunakan penyederhanaan waktu.

-----Halaman ini sengaja dikosongkan-----

BAB V

PENUTUP

Setelah melakukan perencanaan, pembuatan, dan pengujian sistem Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT), maka ditarik kesimpulan dan saran dari kegiatan yang telah dilakukan.

5.1 Kesimpulan

Dari Tugas Akhir yang telah dilakukan, dapat diambil beberapa kesimpulan sebagai berikut.

1. Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT), merupakan bagian dari Prototipe KVARH Meter yang dapat melakukan monitoring energi listrik, baik KWH, KVARH, dan cosphi.
2. Selain memonitoring, juga dapat memberikan notifikasi kepada pelanggan apabila faktor daya kurang dari batas yang sudah ditentukan oleh PLN.
3. Dalam Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT), tidak dapat melakukan fungsi monitoring dan memberikan notifikasi apabila tidak terhubung dengan jaringan internet.
4. Pada pengujian Ethernet Shield, transfer time sangat mempengaruhi pengiriman. Semakin jauh jaraknya, maka semakin lambat pengiriman datanya.
5. Pada pengujian pengiriman data ke Ubidots, data yang dikirim pada Ubidots tingkat ketelitian bilangan desimalnya 4-5 angka di belakang koma. Sedangkan pada serial monitor angka ketelitian bilangan desimalnya hanya 2 angka di belakang koma.
6. Pada pengujian pengiriman notifikasi, data akan dikirim melalui media SMS dan *e-mail* apabila sudah memenuhi kondisi yang sudah ditentukan. Keberhasilan pengiriman notifikasi 100%.
7. Sistem Monitoring dan Notifikasi pada Prototipe KVARH Meter Berbasis *Internet of Things* (IoT), dapat memudahkan pelanggan dalam mengontrol biaya kelistrikannya.

5.2 Saran

Dengan memperhatikan beberapa kelemahan dari proyek tugas akhir ini, maka diberikan beberapa saran yang sekiranya dapat dikembangkan pada masa yang akan datang demi kesempurnaan dari proyek tugas akhir ini. Adapun beberapa saran tersebut yaitu:

1. Pengambilan sampel data, linierisasi data sensor, dan pengujian alat secara keseluruhan lebih banyak dan berulang, untuk memastikan kemampuan alat serta keakuratan alat.
2. Dalam pengujian hendaknya menggunakan alat ukur pembanding digital atau yang lebih akurat agar hasil pengujian bisa mendekati nilai yang sebenarnya.

DAFTAR PUSTAKA

- [1] Pekik Argo D., “Kapasitor: Bermanfaat Sekaligus Berbahaya”, Konversi ITB, Bandung, 2009.
- [2] Nafita H. S., dan Rifki N., "Smart Electricity Meter", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2016.
- [3] Najmi F., dan Fajar W., "Rancang Bangun Kwh Meter Digital yang dilengkapi Data Logger sebagai Alat Bantu melakukan Audit Energi", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2015.
- [4] Ayu Dyah S., "Monitoring Beban KWH Meter 1 Phasa pada pelanggan rumah Tangga dalam Satu Wilayah", *Tugas Akhir*, Institut Teknologi Sepuluh Nopember, Surabaya, 2013.
- [5] Asrul S., ”Rancang Bangun Alat untuk Perbaikan Faktor Daya pada Beban Dinamis 1 Fase dan Monitoring Daya dengan LCD Grafik”, Politeknik Elektronika Negeri Surabaya, , Surabaya, (tanpa tahun).

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN A

A.1 Datasheet Arduino Mega 2560

Features

- High Performance, Low Power Atmel® AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
 - 135 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 × 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16MHz
 - On-Chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 64K/128K/256KBytes of In-System Self-Programmable Flash
 - 4Kbytes EEPROM
 - 8Kbytes Internal SRAM
 - Write/Erase Cycles:10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
 - Endurance: Up to 64Kbytes Optional External Memory Space
- Atmel® QTouch® library support
 - Capacitive touch buttons, sliders and wheels
 - QTouch and QMatrix acquisition
 - Up to 64 sense channels
- JTAG (IEEE® std. 1149.1 compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - Four 16-bit Timer/Counter with Separate Prescaler, Compare- and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four 8-bit PWM Channels
 - Six/Twelve PWM Channels with Programmable Resolution from 2 to 16 Bits (ATmega1281/2561, ATmega640/1280/2560)
 - Output Compare Modulator
 - 8/16-channel, 10-bit ADC (ATmega1281/2561, ATmega640/1280/2560)
 - Two/Four Programmable Serial USART (ATmega1281/2561, ATmega640/1280/2560)
 - Master/Slave SPI Serial Interface
 - Byte Oriented 2-wire Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change

- **Special Microcontroller Features**
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- **I/O and Packages**
 - 54/66 Programmable I/O Lines (ATmega1281/2561, ATmega640/1280/2560)
 - 64-pad QFN/MLF, 64-lead TQFP (ATmega1281/2561)
 - 100-lead TQFP, 100-ball CBGA (ATmega640/1280/2560)
 - RoHS/Fully Green
- **Temperature Range:**
 - -40°C to 85°C Industrial
- **Ultra-Low Power Consumption**
 - Active Mode: 1MHz, 1.8V: 500µA
 - Power-down Mode: 0.1µA at 1.8V
- **Speed Grade:**
 - ATmega640V/ATmega1280V/ATmega1281V:
 - 0 - 4MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega2560V/ATmega2561V:
 - 0 - 2MHz @ 1.8V - 5.5V, 0 - 8MHz @ 2.7V - 5.5V
 - ATmega640/ATmega1280/ATmega1281:
 - 0 - 8MHz @ 2.7V - 5.5V, 0 - 16MHz @ 4.5V - 5.5V
 - ATmega2560/ATmega2561:
 - 0 - 16MHz @ 4.5V - 5.5V

Pin Configurations



Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port A pins that are externally pulled low will source current if the pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B (PB7..PB0)

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port B has better driving capabilities than the other ports.

Port B also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on

Port C (PC7..PC0)

Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D (PD7..PD0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port D also serves the functions of various special features of the ATmega640/1280/1281/2560/2561 as listed on

Port E (PE7..PE0)

Port E is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port E output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port E pins that are externally pulled low will source current if the pull-up resistors are activated. The Port E pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port F (PF7..PF0)

Port F serves as analog inputs to the A/D Converter.

Port F also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port F output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port F pins that are externally pulled low will source current if the pull-up resistors are activated. The Port F pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PF7(TDI), PF5(TMS), and PF4(TCK) will be activated even if a reset occurs.

Port F also serves the functions of the JTAG interface.

Port G (PG5..PG0)

Port G is a 6-bit I/O port with internal pull-up resistors (selected for each bit). The Port G output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port G pins that are externally pulled low will source current if the pull-up resistors are activated. The Port G pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port H (PH7..PH0)

Port H is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port H output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port H pins that are externally pulled low will source current if the pull-up resistors are activated. The Port H pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port J (PJ7..PJ0)

Port J is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port J output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port J pins that are externally pulled low will source current if the pull-up resistors are activated. The Port J pins are tri-stated when a reset condition becomes active, even if the clock is not running. Port J also serves the functions of various special features of the ATmega840/1280/2560 as listed on [page 90](#).

Port K (PK7..PK0)

Port K serves as analog inputs to the A/D Converter.

Port K is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port K output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port K pins that are externally pulled low will source current if the pull-up resistors are activated. The Port K pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Port L (PL7..PL0)

Port L is a 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port L output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port L pins that are externally pulled low will source current if the pull-up resistors are activated. The Port L pins are tri-stated when a reset condition becomes active, even if the clock is not running.

RESET

Reset input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in “[System and Reset Characteristics](#)” on [page 360](#). Shorter pulses are not guaranteed to generate a reset.

XTAL1

Input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting Oscillator amplifier.

AVCC

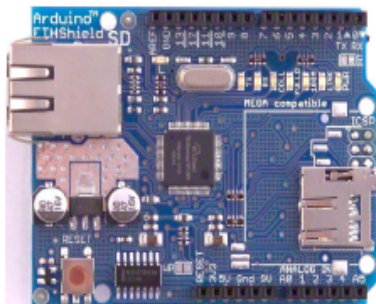
AVCC is the supply voltage pin for Port F and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter.

AREF

This is the analog reference pin for the A/D Converter.

A.2 Datasheet Ethernet Shield

Arduino Ethernet Shield



Download: [arduino-ethernet-shield-05-schematic.pdf](#), [arduino-ethernet-shield-05-reference-design.zip](#)

Download: [arduino-ethernet-shield-schematic.pdf](#), [arduino-ethernet-shield-reference-design.zip](#)

The Arduino Ethernet Shield allows an Arduino board to connect to the internet. It is based on the [Wiznet W5100](#) ethernet chip ([datasheet](#)). The Wiznet W5100 provides a network (IP) stack capable of both TCP and UDP. It supports up to four simultaneous socket connections. Use the [Ethernet library](#) to write sketches which connect to the internet using the shield. The ethernet shield connects to an Arduino board using long wire-wrap headers which extend through the shield. This keeps the pin layout intact and allows another shield to be stacked on top.

The latest revision of the shield adds a micro-SD card slot, which can be used to store files for serving over the network. It is compatible with the Arduino Duemilanove and Mega (using the Ethernet library coming in Arduino 0019). An SD card library is not yet included in the standard Arduino distribution, but the [sdfatlib](#) by Bill Greiman works well. See [this tutorial from Adafruit Industries](#) for instructions (thanks Limor!).

The latest revision of the shield also includes a reset controller, to ensure that the W5100 Ethernet module is properly reset on power-up. Previous revisions of the shield were not compatible with the Mega and need to be manually reset after power-up. The original revision of the shield contained a full-size SD card slot; this is not supported.

Arduino communicates with both the W5100 and SD card using the SPI bus (through the ICSP header). This is on digital pins 11, 12, and 13 on the Duemilanove and pins 50, 51, and 52 on the Mega. On both boards, pin 10 is used to select the W5100 and pin 4 for the SD card. These pins cannot be used for general i/o. On the Mega, the hardware SS pin, 53, is not used to select either the W5100 or the SD card, but it must be kept as an output or the SPI interface won't work.

Note that because the W5100 and SD card share the SPI bus, only one can be active at a time. If you are using both peripherals in your program, this should be taken care of by the corresponding libraries. If you're not using one of the peripherals in your program, however, you'll need to explicitly deselect it. To do this with the SD card, set pin 4 as an output and write a high to it. For the W5100, set digital pin 10 as a high output.

The shield provides a standard RJ45 ethernet jack.

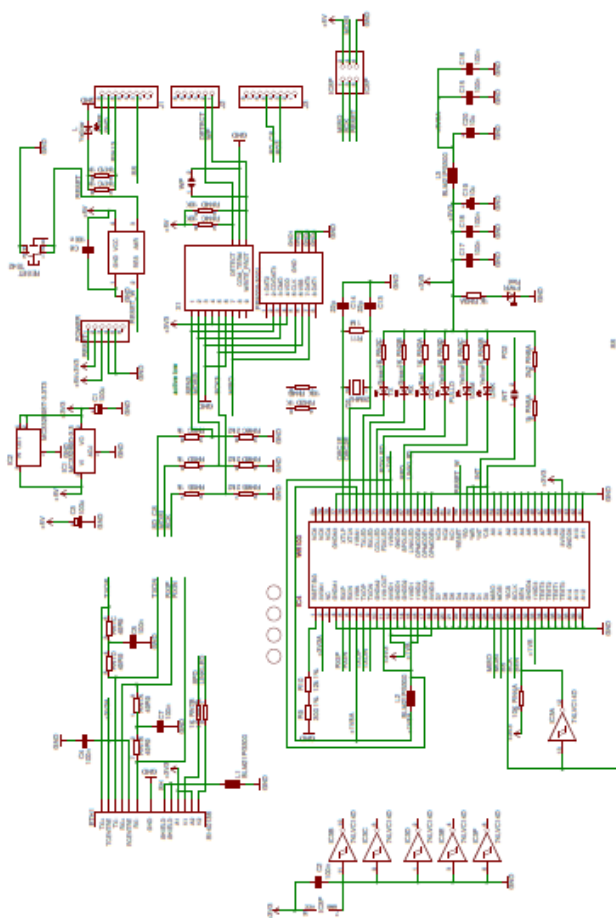
The reset button on the shield resets both the W5100 and the Arduino board.

The shield contains a number of informational LEDs:

- PWR: indicates that the board and shield are powered
- LINK: indicates the presence of a network link and flashes when the shield transmits or receives data
- FULLD: indicates that the network connection is full duplex
- 100M: indicates the presence of a 100 Mb/s network connection (as opposed to 10 Mb/s)
- RX: flashes when the shield receives data
- TX: flashes when the shield sends data
- COLL: flashes when network collisions are detected

The solder jumper marked "INT" can be connected to allow the Arduino board to receive interrupt-driven notification of events from the W5100, but this is not supported by the Ethernet library. The jumper connects the INT pin of the W5100 to digital pin 2 of the Arduino.

See also: [getting started with the ethernet shield](#) and [Ethernet library reference](#)



Arduino ETHERNET - shield V5

Copyright (c) 2010 Arduino
 Released under the Creative Commons Attribution-Share Alike 3.0 License
<http://creativecommons.org/licenses/by-sa/3.0/>

-----Halaman ini sengaja dikosongkan-----

LAMPIRAN B

B.1 Program Keseluruhan

```
#include <Ethernet.h>

#include <SPI.h>

#include <UbidotsEthernet.h>

#define ID1 "596d8a17762542090697f5fa" // Put
here your Ubidots variable ID

#define ID2 "593e3ac37625420179688ab3"

#define ID3 "593e3a7e76254201796886dd"

#define TOKEN "ubsHI8i96Khm77sf5PrR9GHJ3RVYdv"
// Put here your Ubidots TOKEN

#include "EmonLib.h"

#include <LiquidCrystal.h>

#include <DS1307RTC.h>

#include <Wire.h>

#include <Time.h>

#include <DS1307RTC.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

// Include Emon Library

EnergyMonitor emon1;
```

```

// Enter a MAC address for your controller
below.

// Newer Ethernet shields have a MAC address
printed on a sticker on the shield

byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE,
0xED };

// Set the static IP address to use if the DHCP
fails to assign

IPAddress ip(192,168,1,9);


Ubidots client(TOKEN);

void setup()
{
    Serial.begin(9600);

    lcd.begin(16,2);

    emon1.voltage(1, 182.21, 1.7); // Voltage:
input pin, calibration, phase_shift

    emon1.current(3, 14.781);      // Current:
input pin, calibration.

    if (Ethernet.begin(mac) == 0) {

        Serial.println("Failed to configure
Ethernet using DHCP");

        // try to congifure using IP address
instead of DHCP:

        Ethernet.begin(mac, ip);

    }
}

```

```

        // give the Ethernet shield a second to
        initialize:

        delay(1000);

    }

    void loop()

    {

        emon1.calcVI(80,8000);           // Calculate
        all. No.of half wavelengths (crossings), time-
        out

        emon1.serialprint();             // Print out
        all variables (realpower, apparent power, Vrms,
        Irms, power factor)

        float realPower      = emon1.realPower;
        //extract Real Power into variable

        float apparentPower  = emon1.apparentPower;
        //extract Apparent Power into variable

        float powerFactor    = emon1.powerFactor;
        //extract Power Factor into Variable

        float supplyVoltage  = emon1.Vrms;
        //extract Vrms into Variable

        float Irms           = emon1.Irms;
        //extract Irms into Variable

        float Var            =
        sqrt((apparentPower*apparentPower)-
        (realPower*realPower));

```

```

float KWH = KWH + (apparentPower *
(2.05/60/1000));    //Calculate kilowatt hours
used

float KVARH = KVARH + ( Var * (2.05/60/1000));

Serial.print("Arus: ");
Serial.println(Irms);

Serial.print("Tegangan: ");
Serial.println(supplyVoltage);

Serial.print("daya nyata: ");
Serial.println(realPower);

Serial.print("Daya semu: ");
Serial.println(apparentPower);

Serial.print("power factor: ");
Serial.println(powerFACTOR);

Serial.print("Reaktif ");
Serial.println(Var);

Serial.print("KWH :");
Serial.println(KWH);

Serial.print("KVARH: ");
Serial.println(KVARH);

lcd.setCursor(0,0);

lcd.print("KVARH: ");

lcd.print(KVARH);

```

```
lcd.setCursor(0,1);  
lcd.print("cosphi: ");  
lcd.print(powerFactor);  
delay(8000);  
client.add(ID1, powerFactor);  
client.add(ID2, KVARH);  
client.add(ID3, Var);  
    client.sendAll();  
}
```

B.2 Program Mencari IP Address

/*

DHCP-based IP printer

This sketch uses the DHCP extensions to the Ethernet library to get an IP address via DHCP and print the address obtained. using an Arduino Wiznet Ethernet shield.

Circuit:

Ethernet shield attached to pins 10, 11, 12, 13

created 12 April 2011

modified 9 Apr 2012

by Tom Igoe

modified 02 Sept 2015

by Arturo Guadalupi

*/

#include <SPI.h>

#include <Ethernet.h>


```

// Enter a MAC address for your controller below.

// Newer Ethernet shields have a MAC address printed on a sticker on
the shield

byte mac[] = {
  0x00, 0xAA, 0xBB, 0xCC, 0xDE, 0x02
};


// Initialize the Ethernet client library
// with the IP address and port of the server
// that you want to connect to (port 80 is default for HTTP):
EthernetClient client;


void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);

  // this check is only needed on the Leonardo:
  while (!Serial) {

    ; // wait for serial port to connect. Needed for native USB port only
  }


  // start the Ethernet connection:
  if (Ethernet.begin(mac) == 0) {
    Serial.println("Failed to configure Ethernet using DHCP");
  }
}

```

```
// no point in carrying on, so do nothing forevermore:  
  
for (;;)   
  
    ;  
  
}  
  
// print your local IP address:  
printIPAddress();  
  
}
```

```
void loop() {  
  
    switch (Ethernet.maintain())  
    {  
  
        case 1:  
  
            //renewed fail  
  
            Serial.println("Error: renewed fail");  
  
            break;  
  
  
        case 2:  
  
            //renewed success  
  
            Serial.println("Renewed success");  
  
  
  
            //print your local IP address:
```

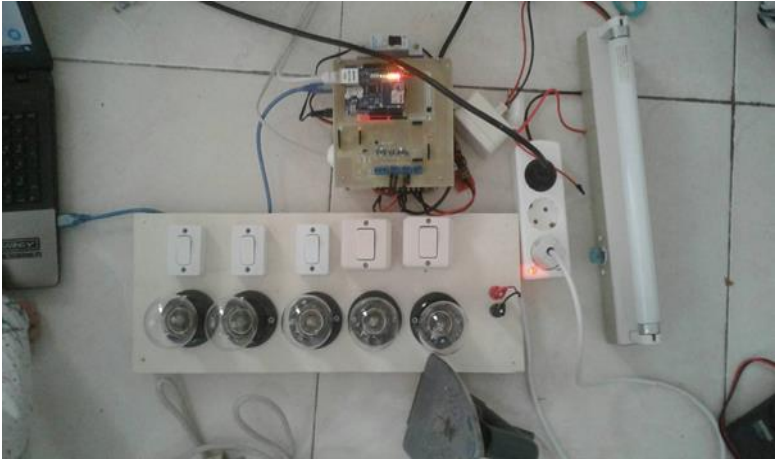
```
    printIPAddress();  
    break;  
  
    case 3:  
        //rebind fail  
        Serial.println("Error: rebind fail");  
        break;  
  
    case 4:  
        //rebind success  
        Serial.println("Rebind success");  
  
        //print your local IP address:  
        printIPAddress();  
        break;  
  
    default:  
        //nothing happened  
        break;  
  
    }  
}
```

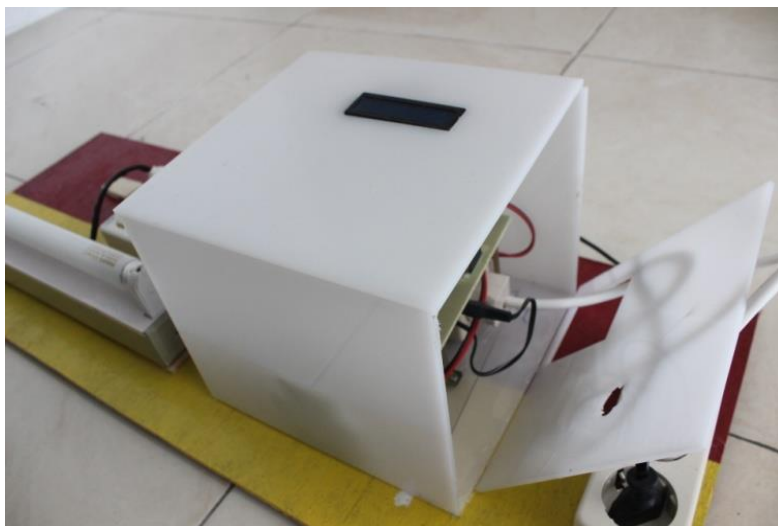
```
void printIPAddress()
{
    Serial.print("My IP address: ");
    for (byte thisByte = 0; thisByte < 4; thisByte++) {
        // print the value of each byte of the IP address:
        Serial.print(Ethernet.localIP()[thisByte], DEC);
        Serial.print(".");
    }

    Serial.println();
}
```

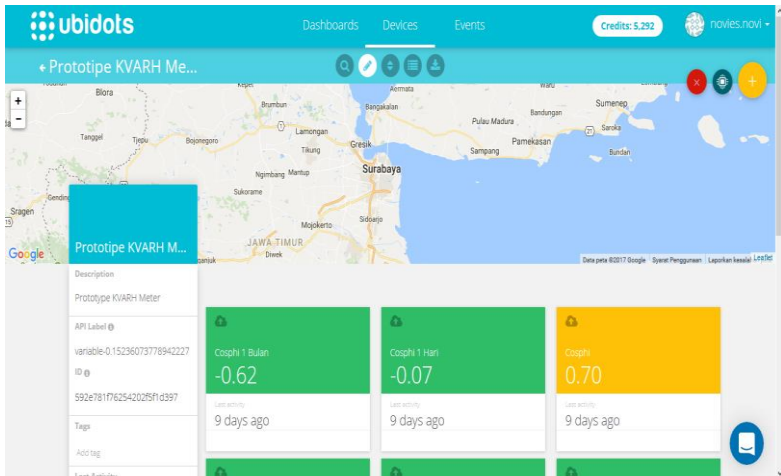
LAMPIRAN C

C.1 Foto Alat






C.2 Tampilan Ubidots



The screenshot shows the Ubidots dashboard interface. At the top, there are tabs for Dashboards, Devices, and Events. The user's name 'novies.novi' and a credit balance of 5,292 are displayed. The main dashboard features a map of Surabaya, Indonesia, with a sidebar showing the device 'Prototype KVARH Me...'. Below the map, there are three data cards:

- Cosphi 1 Bulan**: Value -0.62, Last activity 9 days ago.
- Cosphi 1 Hari**: Value -0.07, Last activity 9 days ago.
- Cosphi**: Value 0.70, Last activity 9 days ago.

ubidots





Dashboard




Devices

Events

My Events

Variable Cosphi 5 Menit is less t...












If Cosphi 1 Hari value is > than -1.0

send sms to +6285607000717

Variable Cosphi 20 Menit is less...












If Cosphi 1 Bulan value is > than -0.85

send sms to +6285607000717

Variable Cosphi 5 Menit is less t...












If Cosphi 1 Hari value is > than -0.85

send email to novies13@gmail.com

Variable Cosphi 20 Menit is less...





If Cosphi 1 Bulan value is > than -0.85

send email to novies13@gmail.com

DAFTAR RIWAYAT HIDUP



Nama : Novi Ainur Riza
TTL : Gresik, 13 November 1996
Jenis Kelamin : Perempuan
Agama : Islam
Alamat : Wadak Kidul 01/01 Duduk
Sampeyan Gresik
No HP : 085607000717
E-mail : novies13@gmail.com

RIWAYAT PENDIDIKAN

1. 2002 – 2008 : MI Infarul Ghoyyi Gresik
2. 2008 – 2011 : MTs Perguruan Mu'allimat Cukir Jombang
3. 2011 – 2014 : SMA A Wahid Hasyim Tebuireng Jombang
4. 2014 – 2017 : Departemen Teknik Elektro Otomasi Fakultas
Vokasi Institut Teknologi Sepuluh Nopember
Surabaya

PENGALAMAN KERJA

1. Magang di PT PLN (Persero) Area Gresik
2. Kerja Praktek di PT PLN (Persero) Rayon Rungkut Area Surabaya Selatan

PENGALAMAN ORGANISASI

1. Wakil Ketua DPM ITS Periode 2015/2016
2. Staf Komisi Legislasi DPM ITS Periode 2014/2015
3. Volunteer BEM ITS Periode 2014/2015
4. Pemandu FTI ITS 2015

-----Halaman ini sengaja dikosongkan-----